MASTER EN SCIENCES INFORMATIQUES

Techniques d'essaimage pour la génération automatique d'un thésaurus

Vangheluwe, Gabriel

Award date:
1975

Awarding institution:
Universite de Namur

FACULTES UNIVERSITAIRES NOTRE-DAME DE LA PAIX A NAMUR
INSTITUT D'INFORMATIQUE

TECHNIQUES D'ESSAIMAGE POUR LA GENERATION AUTOMATIQUE D'UN THESAURUS

ANNEE ACADEMIQUE
1974 - 1975

MEMOIRE PRESENTE PAR
Gabriel VANGHELUWE
POUR L'OBTENTION DU GRADE
DE MAITRE EN INFORMATIQUE

REMERCIEMENTS

Je désire exprimer ma profonde gratitude envers monsieur le professeur J.FICHEFET, pour tous les conseils prodigués; ses nombreuses suggestions et remarques ont permis de clarifier certaines difficultés théoriques, de construire un système expérimental approprié et d'exploiter les résultats obtenus.

C'est avec un grand plaisir que je remercie monsieur le professeur Ph.VANBASTELAER pour l'intérêt porté à ce mémoire; ses critiques m'ont beaucoup aidé à tirer les conclusions relatives à l'indexation des documents.

Qu'il me soit aussi permis de remercier l'équipe du centre de calcul pour leur collaboration durant mes expériences.

Enfin, mes remerciements s'adressent à Mademoiselle S. URTING pour la patience et la bienveillance témoignées tout au long de la frappe de ce mémoire.

TABLE DES MATIERES

RESUME

Nous avons construit des thésaurus de documents et des thésaurus de mots clés à partir d'une collection de documents indexés à l'Institut d'Informatique.Les méthodes d'essaimage qui ont été utilisées sont,pour les thésaurus de mots clés,les cliques maximales,les composantes connexes et la méthode Single-Link,cette dernière méthode servant également pour les thésaurus de documents. Le fonctionnement optimal d'un thésaurus de mots clés correspond à une région du seuil de signification pour lequel le thésaurus est débarrassé des liaisons non significatives,tout en gardant de nombreuses possibilités de substitution.Une étude fréquentielle de la composition du thésaurus montre que la majorité des mots clés du thésaurus n'interviennent qu'une ou deux fois comme descripteurs dans la collection.Le thésaurus contient de nombreuses classes qui ont un noyau commun constitué par des termes de même fréquence et qui se distinguent par un terme supplémentaire,de fréquence supérieure d'une unité.
Les structures de classifications de documents diffèrent considérablement de celles correspondant aux mots clés,les documents se distinguant beaucoup plus les uns des autres.
Enfin,la signification d'un mot clé d'un système de documentation automatique est propre à chaque collection et se déduit de l'ensemble de la collection de documents par sa fréquence d'emploi et ses co-occurrences avec d'autres termes.Une bonne indexation doit être cohérente,non biaisée et complète.Nous avons constaté que les indexeurs ont tendance à ne pas décrire les parties les plus évidentes du contenu des documents,retenant plutôt des détails qui permettent de distinguer entre eux des documents d'un même sujet.

INTRODUCTION

Notre époque est caractérisée par une production sans précédent de publications de toutes sortes:livres,articles,thèses, plans,periodiques,brevets...On estime,par exemple,à plus de 100.000 le nombre de journaux scientifiques connus à ce jour et ce nombre pourrait décupler d'ici l'an 2000. [RESNIKOFF et DOLBY 71], [VICKERY71]

L'un des problèmes majeurs pour un utilisateur de systèmes d'informations consiste à extraire,parmi la masse de publications,tous les documents qui l'intéressent.
La majorité des systèmes de recherche documentaire implémentés permettent d'extraire toutes les informations utiles lorsque l'utilisateur peut exprimer ses besoins de manière précise et complète. Malheureusement,les individus ne possèdent,dans la plupart des cas,qu'une idée plus ou moins intuitive de leurs besoins et la formulation de ceux-ci demeure bien souvent vague,voire même contradictoires. De nombreuses méthodes documentaires ont été proposées en vue d'aider l'utilisateur à préciser ses besoins; citons par exemple:la Classification Décimale Universelle (C D U), les index K W I C (Keyword in Context) et K W I T (Keyword in Text), les fichiers d'auteurs,les fichiers de matières,les dictionnaires de mots clés.
Ces derniers qui donnent les meilleures performances,consistent, le plus souvent,à répertorier et contrôler les termes employés pour indexer les documents et à établir des relations entre ces termes.La construction de ces dictionnaires exige de nombreux efforts et beaucoup de persévérance de la part d'indexeurs spécialisés dans les différentes matières et nécessite souvent des choix arbitraires dans la définition des relations entre les différents termes du vocabulaire admis. Aussi,les recherches se sont-elles orientées vers la conception de procédures automatiques complétant et / ou précisant la requête initiale d'un utilisateur,l'idée finale étant de pouvoir fournir à chacun des renseignements "sur mesure". Les procédures automatiques comportent deux grandes parties:

1 Le traitement de l'information (ou Information Processing) qui comprend notamment les procédures d'analyse de données,les techniques de stockage de l'information, la construction de thésaurus et de dictionnaires de natures diverses.

2 Les stratégies de recherche qui visent à minimiser le nombre de comparaisons (matching) documents-requêtes nécessaires pour énumérer les documents répondant à une requête.

Les techniques d'essaimage sont utilisées à la fois dans la construction de thésaurus et la conception d'algorithmes de recherche.

Elles sont cependant d'un emploi beaucoup plus général. [SIBSON 72] les définit comme: "the study of systems of objects and descriptions with a view to the extraction of information rich - summaries of the original data and possibly the construction of hypotheses within the subject involved".
Elles s'appliquent couramment en biologie, archéologie, psychologie et recherche opérationnelle.
Toutefois, la définition et la construction des essaims demeurent très délicates. [GOWER 69] souligne: " Most of the methods define an algorithm for finding clusters rather than a cluster itself... There is no universal definition of a cluster... Some authors have tried [...] of defining first what they mean by a cluster and then looking for an algorithm for finding clusters with the required properties. Unfortunately, this has so far tended to produce algorithms that are practicable only when clusters are to be found among a few elements. With many elements, approximate methods have to be used, whose properties are as vague as those of the empirical methods".

Les thésaurus constituent la première approche en vue de contrôler le vocabulaire employé lors de l'indexation des documents et la formulation des questions. Leur construction implique deux étapes:

a. Chaque document de la collection est décrit avec un certain nombre de mots ou groupes de mots représentatifs (ou considérés comme tels) du contenu du document.

b. A partir des co-occurences de termes dans la description d'un document, on construit des familles de mots: différentes stratégies de recherche permettent d'exploiter ces dernières en vue de formuler les besoins de l'utilisateur dans le langage d'indexation de la bibliothèque.

Ici, non plus, il n'existe pas de définition rigoureuse de thésaurus: on peut distinguer de nombreuses variantes possibles, selon le but poursuivi. La construction d'un thesaurus procède de l'idée intuitive suivante: [DOYLE 62] :" When pairs of words co-occur strongly in text, it should not be surprising to find the members of the pairs associated in the minds of the authors of books in which they occur". Le thésaurus contient donc une liste d'associations, de co-locations entre termes, construites à partir de relations statistiques, l'objectif étant qu'un vocabulaire structuré doit permettre une plus grande séparation entre documents répondant et ne répondant pas à la question.

La construction de thésaurus pose de nombreux problèmes liés d'une part au nombre de documents à traiter, et d'autre part, à l'évolution dynamique de la collection de documents. En effet, partant d'une collection initiale de documents, il est bien souvent nécessaire de l'adapter graduellement au cours du temps: ajoute de nouveaux documents et de nouveaux mots clés, remplacement de mots clés par plusieurs autres plus spécifiques,

ce qui peut altérer plus ou moins profondément la structure initiale du thesaurus.

Les techniques d'essaimage ont également été utilisées pour résoudre les problèmes de retrieval. Les arguments en faveur de leur emploi sont essentiellement d'ordre économique,car on désire avant tout minimiser le nombre de documents à consulter lors d'une requête tout en veillant à ne pas "oublier" des documents pertinents. Le problème du retrieval comporte deux parties:

1. Construction de la structure du retrieval:elle consiste à former des groupes de documents traitant d'un même sujet et à donner une idée représentative de leur contenu.

2. Consultation de la structure de retrieval:elle consiste à repérer les groupes susceptibles de contenir des documents pertinents,puis à comparer chaque document retenu avec la requête.

Nous verrons que les techniques d'essaimage diffèrent considérablement suivant qu'on les emploie pour construire des thésaurus de mots clés ou,au contraire,pour construire des algorithmes de retrieval.

Les résultats publiés à l'heure actuelle indiquent que l'emploi des thesaurus s'est révélé très bénéfique. Il n'est pas rare,en effet,de voir des améliorations de 40 % des performances par rapport aux implémentationsclassiques procédant par intersections de listes. Il faut toutefois demeurer conscient qu' on ne pourra jamais obtenir une chaîne documentaire parfaite. [SALTON 73] souligne en effet: " The gap between the complete human search and a perfect search [...] appears to be due to ambiguities in query formulations and to the difficulties of reconciling the user's view of a subject area with the subject classification provided in a given document collection. This latter gap may never be bridged by any search and retrieval system likely to come into existence in the foreseeable future".

La construction de thésaurus et d'algorithmes repose donc sur un compromis entre les performances désirées et le coût économique consenti. Celui-ci dépend considérablement du nombre de documents et du nombre de mots clés différents de la collection.

BUT ET IDEES DIRECTRICES DU PRESENT TRAVAIL

Ce mémoire comporte 2 parties.Dans la première à caractère théorique,nous recensons un certain nombre de techniques d'essaimage et nous discutons de leur utilisation en vue d'une application précise : la construction d'un thésaurus et la recherche documentaire basée sur l'exploitation de ce thésaurus.Nous y discutons notamment des problèmes suivants:

1 La construction de matrices de similarité et de graphes à partir du tableau des données d'entrée.

2 Les variantes possibles de définitions d'un essaim et leurs implications sur la structure des thésaurus construits ou sur les algorithmes de retrieval.

3 L'importance relative des différents éléments d'une chaîne documentaire.

4 L'exploitation des renseignements fournis par un utilisateur

5 Les conditions nécessaires pour de bonnes performances de retrieval.

Les problèmes d'occupation mémoire et temps d'exécution sont abordés brièvement.
La deuxième partie expose les expériences effectuées sur un ensemble de documents figurant à l'Institut d'Informatique des FNDP de Namur.Les documents ont été répartis en 3 grandes catégories correspondant plus ou moins à celles établies dans les "Computer Reviews" des ACM,et utilisées dans la bibliothèque de l'Institut:
-Analyse numérique,programmation mathématique,théorie des graphes
-Probabilités,statistiques,processus stochastiques,simulation
-Théorie des langages,Operating Systems,banques de données.
Le but de ces expériences est triple:

1 Construire des thésaurus par les méthodes exposées dans la première partie,à partir des indexations réalisées à l'Institut.

2 Déterminer les paramètres(et leur influence)intervenant dans la construction automatique de thésaurus et dégager des critères permettant,pour une indexation donnée,d'obtenir des performances optimales.

3 Analyser la structure et la composition fréquentielle des thésaurus afin de présenter une étude critique des indexations réalisées et de guider l'indexeur dans l'optique de la construction et de l'exploitation automatiques de thésaurus.

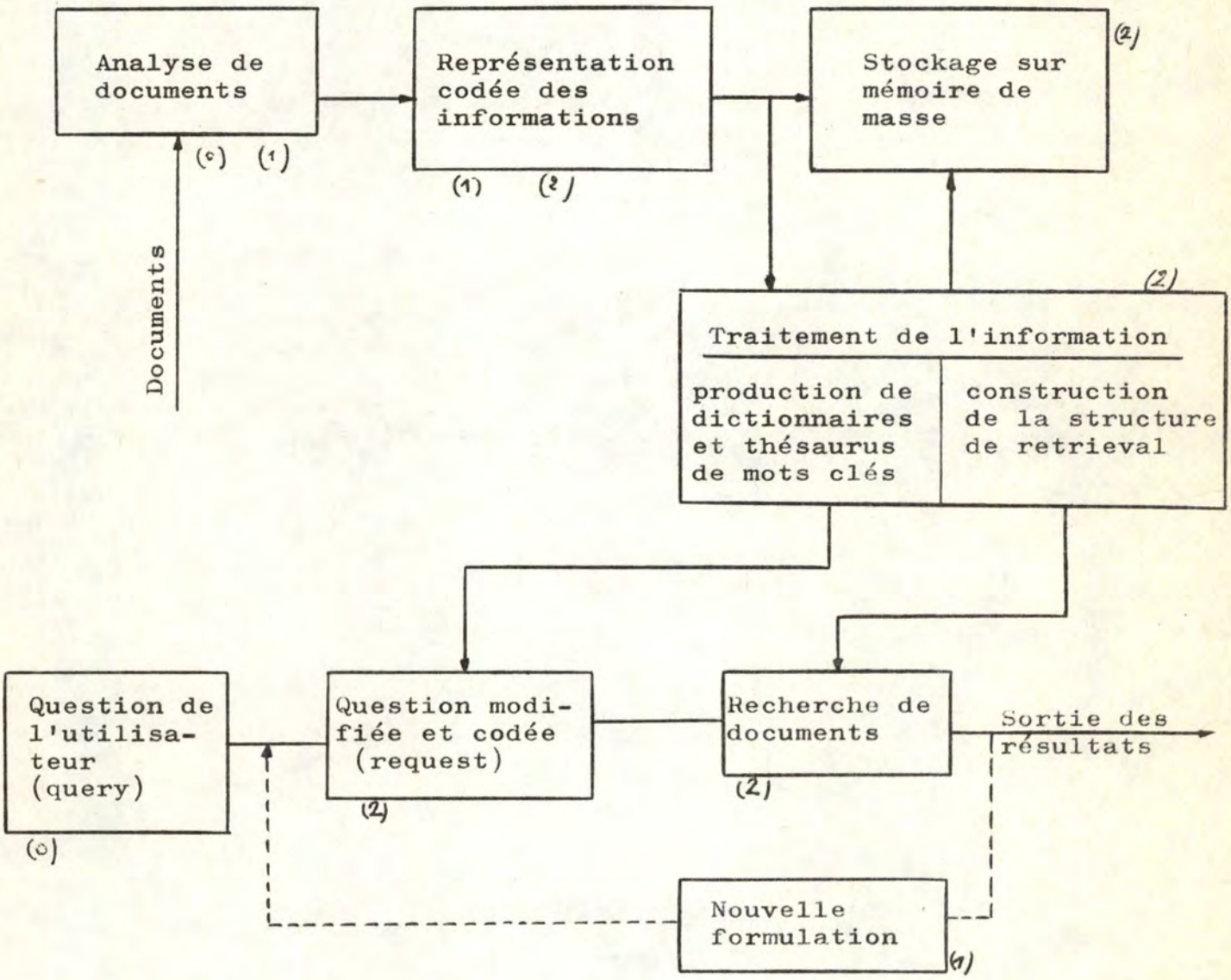
Enfin,nous suggérons quelques projets permettant soit de mieux connaître les différents éléments de la chaîne documentaire,soit d'adapter les techniques proposées à de grosses collections.

CHAPITRE I

GENERALITES

I.1 Définition d'une chaîne documentaire

Nous définissons une chaîne documentaire comme étant le chemin parcouru par l'ensemble des informations depuis leur entrée dans le centre documentaire jusqu'à leur consultation par le lecteur. Une chaîne documentaire se compose des modèles suivants:



(o) procédure manuelle
(1) procédure semi-automatique
(2) procédure automatique

Nous formulons dès à présent quelques remarques très importantes qui conditionnent en fait toute la chaîne documentaire.

1. Les performances de la chaîne documentaire dépendent considérablement du soin avec lequel l'analyse documentaire a été faite:Des traitements informatiques ultérieurs,aussi complexes soient-ils,ne peuvent compenser une analyse de documents par trop rudimentaire.

2. Les performances dépendent également du soin avec lequel l'utilisateur formule sa question:il semble évident que le retrieval ne puisse donner des résultats satisfaisants si la question ne contient que des termes vagues et / ou contradictoires.

3. Le coût de la construction d'une chaîne documentaire est conditionné presque entièrement par le traitement de l'information, c'est-à-dire la construction de thésaurus et de structures de retrieval.
En effet,les modules de traitement font intervenir la plupart du temps,le carré du nombre de mots clés et le carré du nombre de documents.En revanche,une fois consenti cet investissement initial,l'emploi de la chaîne documentaire se révèle très rentable:la consultation de la collection demande en effet,en moyenne log N opérations par question,où N est le nombre de documents.

I.2 Mesure de performance d'une chaîne documentaire

De nombreuses controverses sont nées relativement à la définition de coefficients permettant de mesurer les performances d'une chaîne documentaire.Les deux difficultés principales consistent:

(1)à passer du besoin d'information(état psychologique)à son expression sous forme codée.

(2)à juger de la pertinence d'un document en réponse à une question:deux utilisateurs distincts,formulant la même question,peuvent être intéressés par des documents différents.

A l'heure actuelle,les notions de recall et précision (ou les notions complémentaires de bruit et silence) bien que discutées, servent de standard de mesure.Pour une étude plus approfondie des mesures de performances,on consultera [COOPER 71] , [SALTON et LESK 1969]et [SALTON 71] chapitres 3,4,5] .
Les performances d'une chaîne peuvent être définies au moyen de deux partitions de la collection de documents:
la première est construite par le système de retrieval qui juge tous les documents soit comme répondant,soit comme ne répondant pas à la question (loi du tout ou rien) ;la deuxième partition est du même type mais construite par un staff de personnes dont on suppose qu'elles ne commettent aucune erreur d'appréciation.
En général,les deux partitions seront différentes.On peut alors construire la table ci-dessous:

	Documents repris par le système	Documents non repris par le système
Documents pertinents (repris par un staff de spécialistes)	a	b
Documents non repris par le staff	c	d

Conséquemment, on définit:

Recall $\frac{a}{a+b}$ proportion des documents qui sont énumérés par le système.

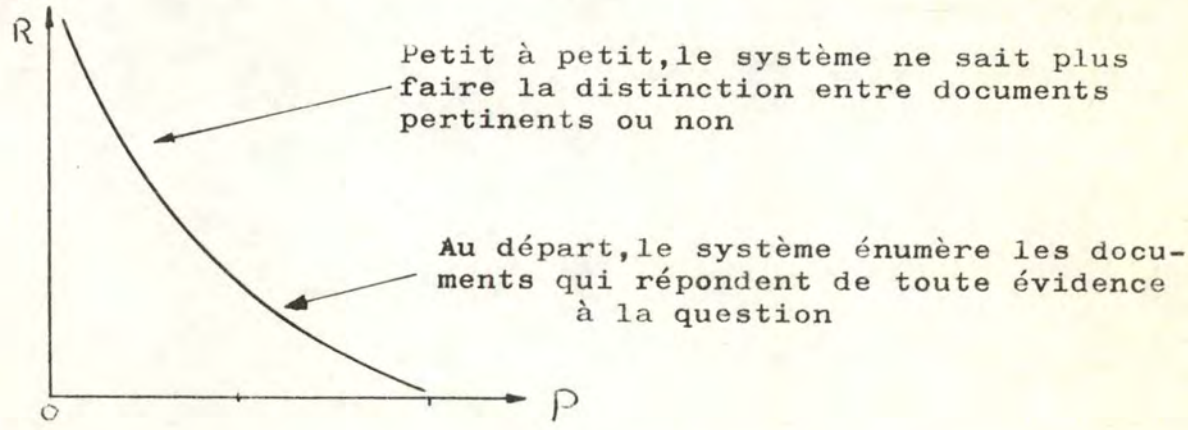
Précision $\frac{a}{a+c}$ proportion des documents énumérés par le système qui sont pertinents.

Bruit $\frac{c}{a+c}$ proportion des documents énumérés par le système qui sont jugés non pertinents.

Silence $\frac{b}{a+b}$ proportion des documents pertinents qui ne sont pas énumérés par le système.

En conclusion, le recall définit l'aptitude du système à ne pas oublier de documents pertinents; au contraire la précision définit l'aptitude du système à ne pas énumérer des documents non pertinents. En d'autres termes, une chaîne de recall 100% fournirait en réponse à une question, tous les documents pertinents (ainsi que les documents non pertinents) alors qu'une chaîne de précision 100% ne fournirait que des documents pertinents (en courant le risque d'en oublier).

En réalité, la définition que nous présentons ici est un peu trop "brutale" et manque de souplesse. Il faut notamment moyenner ces mesures sur de nombreux essais et tenir compte du fait que les documents pertinents ne peuvent être classés par ordre d'importance décroissante. La définition que nous adoptons est toutefois suffisante pour nos besoins.
Un diagramme Recall - Précision présente donc l'allure générale suivante:



[SPARCK - JONES et JACKSON 1970] soulignent l'impossibilité de construire la chaîne documentaire parfaite:" It is true that in a sense,we shoud aim at 100% recall and 100% précision;but this is certainly not attainable in real life,simply because of the character of requests and documents.There must be,that is,some point beyond which we cannot proceed in manipulating requests or document descriptions to optimize their retrieval performance, given the initial character of the descriptions".

Le but de l'analyse de documents est en effet de donner une image codée,significative et concise,mais nécessairement incomplète et unique d'un document en vue de son exploitation ultérieure par l'algorithme de retrieval.
Dès lors,la chaîne documentaire donnera soit une précision soit un recall excellent selon que la description des documents est très détaillée ou non.Cependant,certains artifices tels que la manière de formuler la question,les techniques de pondération des termes,le traitement des termes non significatifs,le paramétrage de l'algorithme de retrieval,son interaction avec l'utilisateur et surtout,le mode de construction du thésaurus,permettent d'articuler plus ou moins la chaîne et donc de modifier les courbes Précision - Recall.

Nous terminerons ce paragraphe en faisant remarquer que les comparaisons avec d'autres chaînes documentaires sont extrêmement difficiles et ce malgré le recours à des standards de mesure.
D'une part,en effet,il peut y avoir des hypothèses et des options tellement différentes concernant les procédures de construction de thésaurus et de retrieval que les courbes Recall - Précision sont difficilement comparables d'une chaîne documentaire à une autre.D'autre part,les diagrammes Recall - Précision,relatifs à une même chaîne documentaire peuvent différer considérablement suivant l'homogénéité des documents de la collection (voir [SPARCK - JONES 1973])

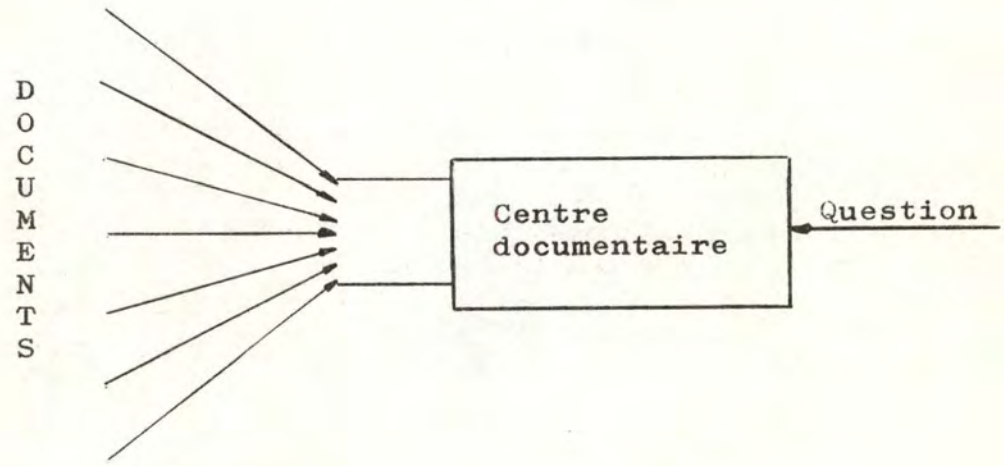
I.3 Influence de l'analyse de documents sur la structure d'un thésaurus

Nous avons déja évoqué au paragraphe précédent quelques problèmes posés par l'analyse de documents. [JACKSON 1971] pose clairement le problème de l'analyse de documents en vue de la construction d'un thésaurus:
"Classification is a process in which the individual differences between the objects to be classified are suppressed in order to derive more general properties which are characteristic of groups of objects.Two questions therefore arise in the present context:

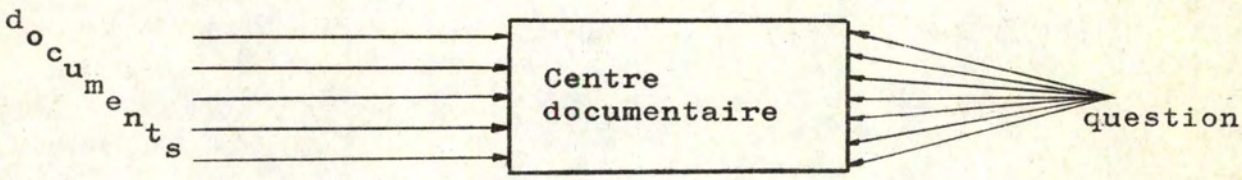
(1) Has too little information been lost,in that the classification permits distinctions to be made between terms which uselessly distinguish between certains documents?

(2) Has too much information been lost,in that the classification does not enable appropriate distinctions to be made between certains documents? "

Dans l'optique thésaurus,l'analyse de documents constitue donc un processus de réduction de données de telle sorte que les classes du thésaurus soient significatives et permettent un retrieval optimal.
[BORKO 64] souligne d'ailleurs le danger de descriptions trop précises: " One of the claims for mathematically derived classification systems is that the categories so derived are descriptions of the documents used in the analysis.However,if the categories prove to be so unique that they describe only the one document set and no other,they would be of little value".
Le cas limite dans lequel il n'y a aucune réduction de données, a été traité par [SWANSON en 1963] ;mais cette fois,c'est l'utilisateur qui doit résoudre le problème de l'amplification de la question et l'effort exigé pour formuler ses besoins est beaucoup plus considérable.(voir ci-dessous,l'exemple tiré de [CROS-GARDIN-LEVY])



Processus de réduction de l'information initiale: un document caractérisé par un ensemble de mots clés ; de plus,la réduction de l'information initiale s'opère sur des documents concrets. Une question est formulée au moyen de quelques mots clés.



Processus d'amplification de la question:les documents sont stockés tels quels sur mémoire de masse. L'utilisateur doit décrire des documents virtuels.

CHAPITRE II

LES TECHNIQUES D'ESSAIMAGE.

II.1 Introduction.

Les techniques d'essaimage(clustering techniques ou cluster analysis)sont abondamment employées depuis une quinzaine d'années, comme moyen de classification (+).Les problèmes de classification apparaissent sous des formes très variées dans des domaines aussi divers que la biologie(classification de plantes ou de bactéries: voir la bibliographie dans [MAC CORMACK 71]),l'archéologie(classification de tombes étrusques[WILKINSON 71]),la construction de circuits électroniques(en vue de minimiser la longueur des connexions entre composants électroniques[BREUER 72]),la recherche documentaire,la propagation d'épidémies(PIKE et SMITH 68,PIKE et PETO 70), la recherche opérationnelle(problème du voyageur de commerce,affectation de ressources -personnel et matériel- à un certain nombre d'activités en vue de minimiser les coûts de transfert d'une activité à une autre[WARD 63,GREEN,FRANK,ROBINSON 68,JENSEN 68,MAC NAUGHTON-SMITH 63])...
Tous ces problèmes possèdent cependant la même structure de données, à savoir un ensemble P d'objets(tombes,bactéries,documents,...) et un ensemble E de caractères décrivant ces objets(pour les tombes étrusques:poteries,armes,bijoux,peintures,...).Ces données peuvent prendre la forme d'un tableau rectangulaire N×M où les lignes représentent les objets et les colonnes,les caractéristiques des objets($N=|P|$ et $M=|E|$).
Les entrées dans la matrice doivent être de même type à l'intérieur d'une même colonne,mais peuvent être,par ailleurs,différentes dans une même matrice.Les objets peuvent être décrits par des attributs, des données numériques,des données ordinales,des distributions de probabilité...
L'objectif des techniques d'essaimage est d'établir une classification des objets suivant leurs caractères.On peut également résoudre le problème inverse:classer les caractères suivant leur appartenance aux différents objets:c'est le cas des classifications de mots clés dans un système de documentation automatique.

Les méthodes pratiques d'analyse des matrices objets-caractères comparent un nombre restreint d'objets entre eux,généralement deux;les résultats de ces comparaisons deux à deux sont exprimés sous forme d'une valeur numérique supposée représenter la similarité(ou dissimilarité)entre 2 objets.Après application de ce processus,on perd toute information concernant les relations entre caractères et les relations entre objets et caractères.

+ Intuitivement,un essaim est un ensemble d'objets caractérisé par des propriétés de cohérence et d'isolement.

La fonction f:P×P→R,au moyen de laquelle on compare les objets est appelée fonction de comparaison,ou encore coéfficient de dissimilarité(ou similarité),parfois aussi,coéfficient de corrélation:elle se présente sous forme d'une matrice N×N dont les lignes et les colonnes sont les éléments de P.Cette matrice de comparaison est le point de départ de la plupart des classifications:pour d'autres méthodes,voir[MAC CORMACK 71] et[JENSEN 68],ou l'algorithme de DATTOLA.

Les classifications diffèrent par:

-Le choix des fonctions de comparaison,lui-même conditionné par les types de données caractérisant les objets et l'importance relative accordée aux différents caractères.

-L'usage de la classification:selon le but poursuivi,on construira:

a. Des classes hiérarchisées ou non:dans une classification hiérarchique,les classes se subdivisent en groupes, le processus se répétant à différents niveaux.

b. Des classes mutuellement exclusives ou non:suivant que l'on désire partitionner l'ensemble des objets,ou que l'on permet à un objet de figurer dans plusieurs classes.

On peut néanmoins dégager quelques propriétés générales que tout algorithme d'essaimage devrait vérifier:

P1 L'essaimage doit être significatif:l'algorithme doit fournir des essaims correspondant à une structure effectivement possédée par les données(un problème analogue se pose en statistique:la moyenne par exemple,n'est pas significative d'une distribution bi-modale).

P2 Si les données possèdent effectivement la structure recherchée, la technique d'essaimage ne doit pas faire apparaître une autre structure:en d'autres termes,la méthode d'essaimage doit être adaptée au but poursuivi.

P3 La méthode doit être indépendante de toute permutation dans la numérotation des objets.

P4 L'essaimage doit être indépendant de tout facteur d'échelle:la procédure doit fournir les mêmes résultats si on multiplie la matrice de comparaison par une constante strictement positive.

P5 La procédure d'essaimage doit être stable vis-à-vis

a. de légères modifications des données d'entrée,afin que la méthode soit insensible aux erreurs dans les données d'entrée.

b. d'ajoutes et suppressions de données:Les essaims ne doivent pas être complètement modifiés par l'ajoute de quelques objets ou caractères.

Soulignons également que les méthodes pratiques d'essaimage doivent pouvoir s'appliquer avec des temps d'exécution raisonnables,à des collections comportant plusieurs centaines ou milliers d'objets:

ainsi,les classifications hiérarchiques non exclusives,qui sont les plus riches du point de vue contenu d'information ne peuvent être envisagées au delà de N=50 et sont donc d'intérêt limité. Nous présentons ci-dessous un tableau donnant les techniques d' essaimage que nous allons exposer.Nous indiquons également les propriétés que chaque méthode possède;nous n'avons pas étudié les propriétés P2 et P5,car elles sont liées aux buts poursuivis et aux données présentées à l'algorithme.

	P_1	P_3	P_4
Single-Link	1	1	1
composantes connexes	1	1	1
cliques maximales	1	1	1
étoiles		1	1
aggrégats de NEEDHAM	1		1
algorithme de ROCCHIO			1
algorithme de DATTOLA			1

II.2 Définitions

II.2.1. Rappel

Un certain nombre d'algorithmes opèrent sur un graphe construit à partir de la matrice de comparaison;c'est pourquoi,nous rappelons préalablement quelques définitions de la théorie des graphes:

1 Graphe_non_orienté_ :est défini par un ensemble de sommets X_i et un ensemble d'arêtes X_i, X_j

Composante_connexe d'un graphe non orienté:l'ensemble des sommets d'un graphe tel qu'il existe une chaîne reliant 2 sommets quelconques de cet ensemble constitue une composante connexe Les composantes connexes forment une partition des sommets du graphe.

Clique : L'ensemble des sommets d'un graphe tel que 2 sommets quelconques de cet ensemble sont reliés par une arête constitue une clique.Une clique construite sur un graphe est maximale si elle n'est contenue dans aucune autre clique construite sur ce graphe.

Dendrogramme : Un dendrogramme est un cas particulier d'un arbre dans lequel on a associé des niveaux numériques aux différents noeuds.Les essaims correspondant à un niveau particulier dans le dendrogramme forment une partition de l'ensemble P des objets.Ces niveaux numériques permettent d'établir une hiérarchie dans les liaisons entre objets ou

groupes d'objets:au niveau le plus bas ne sont reliés entre eux que des objets quasi identiques;au niveau le plus élevé,il n'y a plus de distinction entre les différents objets.Une définition précise d'un dendrogramme est donnée au paragraphe II.3.1.

II.2.2 Fonctions de comparaison.

Une énumération des principales fonctions de comparaison est faite dans [BALL 65], [LERMAN 70], [MAC CORMACK 71] .Des études théoriques et expérimentales ont été entreprises en vue de déterminer l'influence du choix d'une fonction de comparaison sur les essaims obtenus (voir par exemple [LERMAN 70],[SPARCK-JONES et JACKSON 70]).Nous nous contenterons de reprendre les conclusions de [HARRISON 68]: "Generally,in cases where it is relatively easy to form clusters using the values of a given function,it is unlikely that substantially different clusters will be obtained by using the values of a different function:if on the other hand the clustering process is difficult,and the information does not appear to lead naturally to compact clusters,it is unlikely that the selection of any particular function will make the process casier,unless substantial loss of information has taken place".

Nous exposons ci-dessous les caractéristiques générales des fonctions de comparaison.Soit P un ensemble d'objets;chaque objet est représenté par un vecteur-ligne de la matrice objets-caractères.

	C_1	C_2	C_3	C_4	C_5	C_6
a_1	1	0	1	0	1	0
a_2	1	1	0	0	0	1
a_3	0	1	1	0	1	1
a_4	1	0	1	0	1	0

exemple de matrice objet-caractère (à données binaires):les objets sont représentés sous forme de vecteurs-lignes et les caractères sous forme de vecteurs-colonnes.

Soit de plus,d une fonction de comparaison,définie sur P×P,à valeurs réelles;d satisfait au moins aux conditions suivantes

(1) $d(a_1,a_2) \geq 0 \quad \forall\ a_1,a_2 \in P$

(2) $d(a_1,a_1) = 0 \quad \forall\ a_1 \in P$

(3) $d(a_1,a_2) = d(a_2,a_1) \quad \forall a_1,a_2 \in P$

On impose souvent d'autres contraintes,et notamment

(a) $d(a_1,a_2) = 0 \Rightarrow d(a_1,a_3) = d(a_2,a_3) \ \forall a_1,a_2,a_3 \in P$

Cette relation exprime que 2 objets distincts(par exemple, un même objet compté 2 fois)doivent se trouver à la même "distance" de tous les autres objets.

(b) $d(a_1,a_2) = 0 \Rightarrow a_1 = a_2 \quad \forall a_1,a_2 \in P$

Si 2 objets sont dissemblables et d'une quantité nulle,ils sont identiques.

(c) $d(a_1,a_2) \leq d(a_1,a_3) + d(a_3,a_2) \quad \forall a_1,a_2,a_3 \in P$
(inégalité triangulaire)

(d) $\max\{d(a_1,a_2), d(a_2,a_3)\} \geq d(a_1,a_3) \quad \forall a_1,a_2,a_3 \in P$
(inégal. ultramétrique)

On peut montrer que (d)$\Rightarrow$(c)$\Rightarrow$(a)$\Leftarrow$(b).
On remarquera que si les conditions (1),(2),(3),(b),et(c) sont satisfaites,la fonction d est une distance sur P.Toute distance sur P est d'ailleurs une fonction de comparaison,mais la réciproque est généralement fausse.

Nous illustrons la notion de fonction de comparaison au moyen de la matrice documents-termes N x M présentée ci-dessus(N = nombre de documents,M = nombre de termes).Les matrices présentées sont les matrices de similarité(dont les éléments sont normalisés entre 0 et 1)documents-documents(N x N) et termes-termes(M x M).
On sait que si $d(i,j) \in [0,1]$ mesure la dissimilarité et $S(i,j) \in [0,1]$ la similarité entre 2 objets i et j,alors $S(i,j) = 1-d(i,j)$

Les mesures de similarité employées sont

la mesure de Tanimoto $S(a_1,a_2) = \dfrac{\sum_j^N a_1^{(j)} \times a_2^{(j)}}{\sum_j^N a_1^{(j)} + \sum_j^N a_2^{(j)} - \sum_j^N a_1^{(j)} a_2^{(j)}}$

la mesure cosinusoïdale $s(a_1,a_2) = \dfrac{\sum_j^N a_1^{(j)} \times a_2^{(j)}}{\left[\sum_j^N \left(a_1^{(j)}\right)^2 \times \left(a_2^{(j)}\right)^2\right]^{1/2}}$

	a_1	a_2	a_3	a_4
a_1	1			
a_2	$\frac{1}{5}$	1		
a_3	$\frac{2}{5}$	$\frac{2}{5}$	1	
a_4	$\frac{3}{5}$	$\frac{1}{5}$	$\frac{2}{5}$	1

Matrice de similarité doc-doc; mesure de TANIMOTO

	a_1	a_2	a_3	a_4
a_1	1			
a_2	$\frac{1}{\sqrt{3\times3}}$	1		
a_3	$\frac{2}{\sqrt{3\times4}}$	$\frac{2}{\sqrt{3\times4}}$	1	
a_4	$\frac{3}{\sqrt{3\times3}}$	$\frac{1}{\sqrt{3\times3}}$	$\frac{2}{\sqrt{3\times4}}$	1

Matrice de similarité doc-doc; mesure cosinusoïdale

	C_1	C_2	C_3	C_4	C_5	C_6
C_1	1					
C_2	$\frac{1}{4}$	1				
C_3	$\frac{2}{4}$	$\frac{1}{4}$	1			
C_4	0	0	0	1		
C_5	$\frac{2}{4}$	$\frac{1}{4}$	$\frac{3}{3}$	0	1	
C_6	$\frac{1}{4}$	$\frac{2}{2}$	$\frac{1}{4}$	0	$\frac{1}{4}$	1

Marice de similarité termes-termes; mesure de TANIMOTO

	C_1	C_2	C_3	C_4	C_5	C_6
C_1	1					
C_2	$\frac{1}{\sqrt{3\times2}}$	1				
C_3	$\frac{2}{\sqrt{3\times3}}$	$\frac{1}{\sqrt{2\times3}}$	1			
C_4	0	0	0	1		
C_5	$\frac{2}{\sqrt{3\times3}}$	$\frac{1}{\sqrt{2\times3}}$	$\frac{3}{\sqrt{3\times3}}$	0	1	
C_6	$\frac{1}{\sqrt{3\times2}}$	$\frac{2}{\sqrt{2\times2}}$	$\frac{1}{\sqrt{3\times2}}$	0	$\frac{1}{\sqrt{3\times2}}$	1

Matrice de similarité termes-termes; mesure cosinusoïdale

On remarquera qu'une matrice de similarité(ou de dissimilarité) objets-objets(caractères-caractères)nécessite au moins $\frac{N(N-1)}{2}$ comparaisons vectorielles où N = nombre d'objets(de caractères). Le temps d'exécution est donc au moins $O(N^2)$.

II.3 Algorithmes d'essaimage

II.3.1 La méthode SINGLE-LINK

La méthode SINGLE-LINK a pour but de construire un dendrogramme. Nous savons que si r est en relation d'équivalence sur P(l'ensemble des objets,alors les ensembles $C_\alpha = \{\beta : (\alpha,\beta) \in r\}$ forment une partition de P;réciproquement,si les ensembles $\{C_\gamma\}$ forment une partition de P,alors $r = \bigcup_\alpha (C_\alpha \times C_\alpha)$ est une relation d'équivalence sur P.Les dendrogrammes ayant pour but de construire des partitions imbriquées,nous pouvons utiliser une relation d'équivalence pour caractériser un dendrogramme.Nous devons également tenir compte des niveaux numériques du dendrogramme:si $h < h'$ et si C est un essaim au niveau h,alors C est inclus dans un essaim C' au niveau h';pour h suffisamment grand,il n'y a qu'un seul essaim.

On imposera en outre une condition de stabilité des essaims en passant d'un niveau au suivant. Une définition formelle d'un dendrogramme pourrait donc être celle-ci:

Soit E(P) l'ensemble des relations d'équivalence sur P.
Un dendrogramme est une fonction C: $[0,\infty[\rightarrow E(P)$ satisfaisant aux conditions suivantes:

(1) $0 \leq h \leq h' \Rightarrow C(h) \subseteq C(h')$

(2) $C(h) = P \times P$ pour h suffisamment grand

(3) $C(h + \delta) = C(h)$ pour δ suffisamment petit

La relation d'équivalence utilisée par la méthode SINGLE-LINK est définie comme suit: soit d le coéfficient de dissimilarité choisi, permettant de comparer les objets 2 à 2. Pour une valeur fixée de h, considérons le graphe dont les sommets sont les objets à essaimer et dont les arêtes joignent les paires de sommets (i,j) pour lesquels $d(i,j) \leq h$. Alors C(h) est la relation d'équivalence correspondant à la partition de P définie par les composantes connexes de ce graphe.
L'algorithme est celui de [SIBSON 73] considéré comme le plus efficace à l'heure actuelle.
Nous définissons préalablement une représentation par pointeurs comme étant une paire de fonctions $\pi : \{1,\ldots,N\} \rightarrow \{1,\ldots,N\}$ et $\lambda : \{1,\ldots,N\} \rightarrow [0,\infty]$ satisfaisant aux conditions suivantes: (N représente le nombre d'objets à essaimer)

$$\pi(N) = N \quad ; \quad \lambda(N) = \infty \qquad (1)$$

$$\pi(i) > i \quad ; \quad \lambda(\pi(i)) > \lambda(i) \quad \text{pour } i < N \qquad (2)$$

Comme l'algorithme de construction d'un dendrogramme opère sur (π,λ), nous allons montrer que les transformations $C \rightarrow (\pi,\lambda)$ et $(\pi,\lambda) \rightarrow C$ sont mutuellement inverses.
Nous énonçons auparavant 2 propositions:

1 Etant donné un dendrogramme C, on peut en déduire une représentation par pointeurs π, λ de la manière suivante

$$\lambda(i) = \inf \{ h : \exists j > i \text{ avec } (i,j) \in C(h) \} \qquad i < N \qquad (3a)$$

$$\pi(i) = \max \{ j : (i,j) \; C(\lambda(i)) \} \qquad i < N \qquad (3b)$$

$\lambda(i)$ est donc la plus petite valeur de h pour laquelle il existe un sommet de numéro $> i$ figurant dans le même essaim que i; $\pi(i)$ mémorise le sommet auquel correspond le plus grand numéro figurant dans le même essaim que i au niveau $\lambda(i)$.
Le couple (π,λ) est une représentation par pointeurs: en effet, il existe nécessairement, pour $i < N$, un $j > i$ et un h tel que $(i,j) \in C(h)$, ce qui permet de fixer $\lambda(i)$ et $\pi(i)$

2 Réciproquement, étant donné une représentation par pointeurs (définie par (1), (2)), on peut en déduire un dendrogramme.
Soit (π,λ) une représentation par pointeurs et la suite (strictement croissante) $i, \pi(i), \pi^2(i), \ldots, N$ (4)

On définit une fonction $\tau(i,h)$ en prenant pour $\tau(i,h)$ le premier élément k de la suite (4) tel que $\lambda(k) > h$.

Dès lors $C(h) = \{(i,j) : \tau(i,h) = \tau(j,h)\}$ est un dendrogramme.

Démonstration

a. $h \leq h' \Rightarrow C(h) \subseteq C(h')$
En effet, si $h \leq h'$ alors $\tau(i,h) \leq \tau(i,h')$

$\tau(j,h) \leq \tau(j,h')$

Or, si $(i,j) \in C(h)$, on a également $(m,n) \in C(h)$
avec $m \in \{i, \pi(i), \pi^2(i), \ldots, \tau(i,h)\}$
$n \in \{j, \pi(j), \pi^2(j), \ldots, \tau(j,h)\}$

On peut faire de même pour $C(h')$
Dès lors, $(m,n) \in C(h) \Rightarrow (m,n) \in C(h')$

b. Les conditions (2) et (3) de la définition d'un dendrogramme sont également vérifiées.

Théorème: Les transformations $C \rightarrow (\pi,\lambda)$ et $(\pi,\lambda) \rightarrow C$ sont mutuellement inverses.

a. Démontrons que $C \rightarrow (\pi,\lambda) \rightarrow C'$ entraîne $C' = C$

Il suffit de démontrer $C(h) \subseteq C'(h)$ et $C'(h) \subseteq C(h)$

A partir de (π,λ), on peut construire le dendrogramme

$C'(h) = \{(i,j) ; \tau(i,h) = \tau(j,h)\}$

Mais (π,λ) est déduit de $C \Rightarrow (i, (i,h)) \in C(h)$

$(j, (j,h)) \in C(h)$

Comme $\tau(i,h) = \tau(j,h) \Rightarrow (i,j) \in C(h)$

Donc $C'(h) \in C(h)$

De même,
si $(i,j) \in C(h) \Rightarrow (\tau(i,h), \tau(j,h)) \in C(h)$

supposons $\tau(i,h) \neq \tau(j,h)$: par exemple

$\tau(i,h) < \tau(j,h)$

Or, par définition,

$\lambda(i_1) = \inf \{h_1 : \exists\, j_1 > i_1$ avec $(i_1,j_1) \in C(h_1)\}$

Nous voyons que l'inf est réalisée ici pour:

$h_1 = h, i_1 = \tau(i,h), j_1 = \tau(j,h)$

Dès lors, $\lambda(\tau(i,h)) \leq h$ ce qui est faux par construction de $\tau(i,h) \Rightarrow \tau(i,h) = \tau(j,h)$

Donc $(i,j) \in C(h) \Rightarrow (i,j) \in C'(h)$

b. Démontrons que $(\pi,\lambda) \rightarrow C \rightarrow (\pi',\lambda')$ entraîne $\pi' = \pi$ et $\lambda' = \lambda$

C étant un dendrogramme, nous en déduisons

$\lambda'(i) = \inf \{h : \exists\, j > i$ avec $(i,j) \in C(h)\}$

$= \inf \{h : \exists\, j > i$ avec $\tau(i,h) = \tau(j,h)\}$

Or on sait que $(i, \tau(i,h)) \in C(h)$; dès lors, la condition précédente peut s'exprimer en remplaçant j par $\tau(i,h)$

De même, $\pi'(i) = \max \{ j : (i,j) \in C(\lambda'(i)) \}$

$= \max \{ j : (i,j) \quad C(\lambda(i)) \}$

$= \max \{ j : \sigma(i,\lambda(i)) = \sigma(j,\lambda(i)) \}$

$= \max \{ j : \pi(i) = \sigma(j,\lambda(i)) \}$

$= \pi(i)$

Nous illustrons la représentation par pointeurs sur le dendrogramme ci- dessous

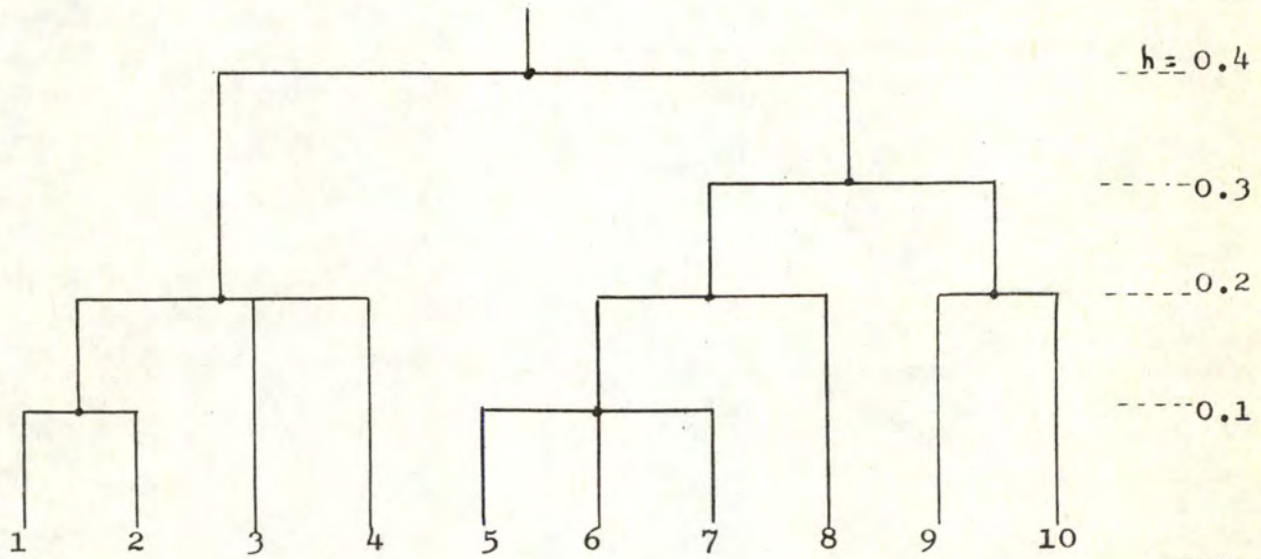


FIG II.1

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
π	2	4	4	10	7	7	8	10	10	10
λ	.1	.2	.2	.4	.1	.1	.2	.3	.2	∞

L'algorithme de SIBSON procède par itérations successives:à chaque itération,il lit une ligne de la matrice triangulaire inférieure des D.C.(dissimilarity coefficient),étend le dendrogramme existant avec le nouvel objet ajouté,correspondant au n° de la ligne lue, et met à jour les vecteurs (η, λ) de telle sorte que les (π, λ) ainsi modifiés soient la représentation du nouveau dendrogramme.
Le passage de l'itération n→ n+1 se fait comme suit:soit m le vecteur qui contiendra la n+1ème ligne de la matrice.

1 $\pi(n+1) \leftarrow n+1$; $\lambda(n+1) \leftarrow \infty$

2 $m(i) \leftarrow DC(i,n+1)$ pour $i \in [1,n]$

3 pour i allant de 1→ n

$\underline{si}\ \lambda(i) \geq m(i)\ \underline{alors}\ m(\pi(i)) \leftarrow \min\{ m(\pi(i)), \lambda(i) \}$

$\lambda(i) \leftarrow m(i)$

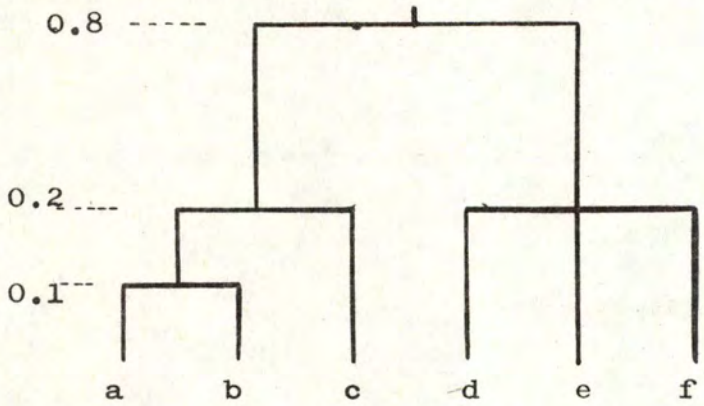
$\pi(i) \leftarrow n+1$

$\underline{sinon}\ m(\pi(i)) \leftarrow \min\{ m(\pi(i)), m(i) \}$

4 pour i allant de 1 → n

$\underline{si}\ \lambda(i) \geq \lambda(\pi(i))\ \underline{alors}\ \pi(i) \leftarrow n+1$

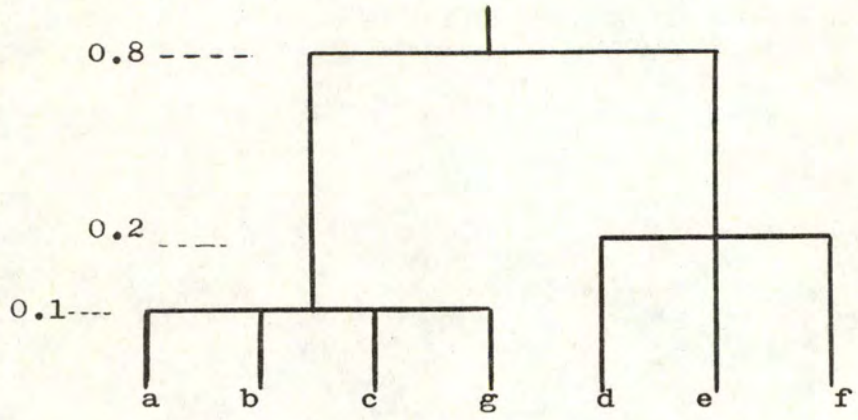
Une justification rigoureuse de la procédure employée est formulée dans [SIBSON 73] .Nous nous contenterons ici de présenter un exemple d'évolution du dendrogramme au cours d'itérations pour quelques données d'entrée particulières.



Supposons que la lecture de la ligne n° g donne notamment DC(g,b) = 0.1

DC(g,c) = 0.1

FIG II.3



La lecture de la ligne h donne entre autres:

DC(h,c) = 0.6

DC(h,f) = 0.6

FIG II.3

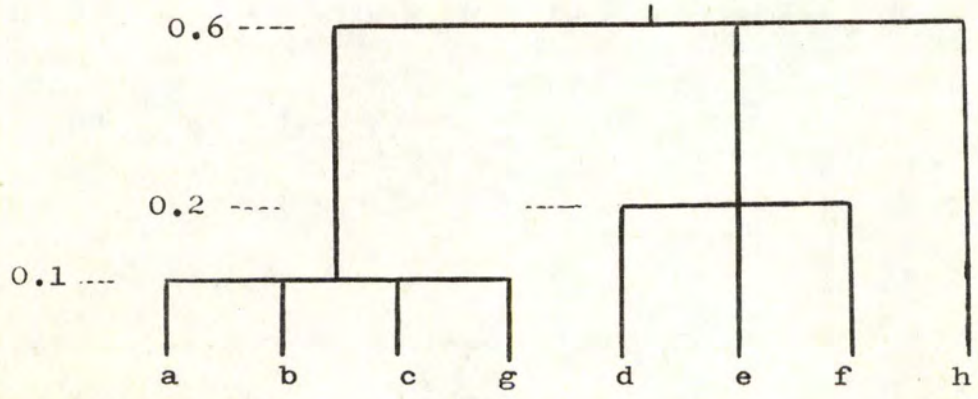


FIG II.4

La méthode ne requiert que 3N mots mémoires(alors que la plupart des autres méthodes nécessitent des accès aléatoires à un élément quelconque de la matrice de DC ce qui exige son implantation en mémoire centrale).D'autre part,le nombre d'opérations pour la mise à jour des pointeurs est $O(N^2)$,et la procédure employée à chaque itération est très simple: La méthode est donc très efficace.

Généralement,on transforme cette représentation en d'autres plus maniables:

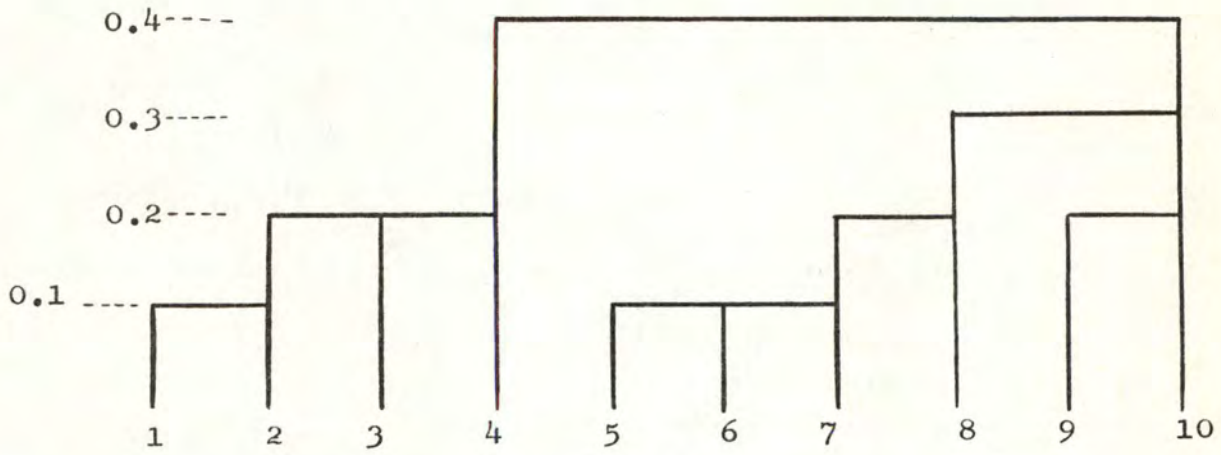
1 La transformation$(\pi,\lambda)\rightarrow(y,h)$ afin de permettre le tracé du dendrogramme.

y(i) représente le n° de l'objet figurant en $i$ème position dans le dendrogramme.

h(i) représente la hauteur du ième trait vertical.
Le tracé du dendrogramme est alors immédiat:

a Tracer les barres verticales dont la hauteur est donnée par le vecteur h.

b A partir du sommet de chaque trait vertical,tracer une barre horizontale,jusqu'à ce qu'on rencontre une barre verticale.Le tracé du dendrogramme de la figure II.1 serait donc:



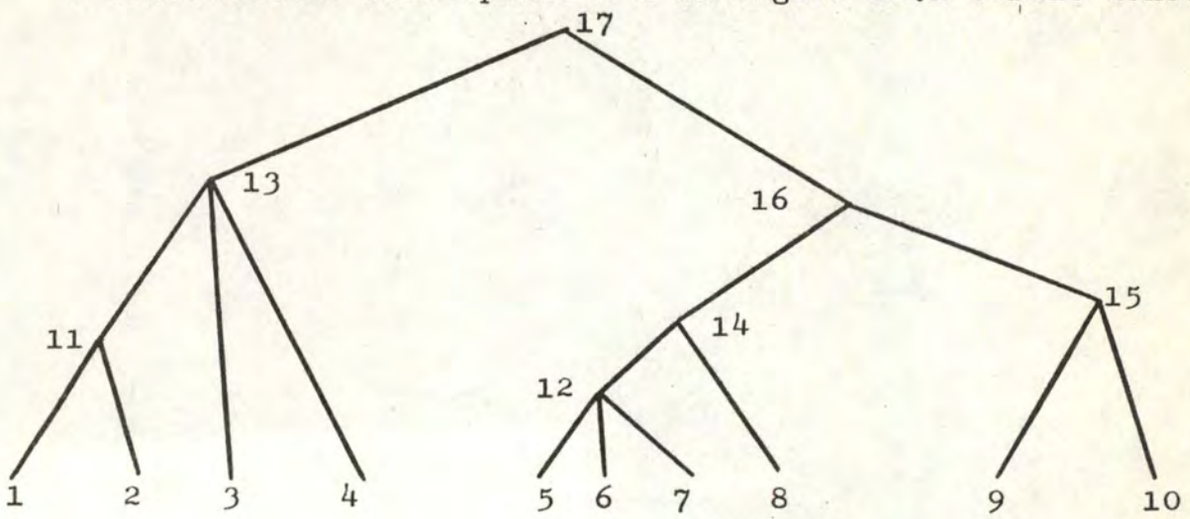
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
y	1	2	3	4	5	6	7	8	9	10
h	.1	.2	.2	.4	.1	.1	.2	.3	.2	.5

2 La transformation$(\pi,\lambda)\rightarrow$(frère,fils) qui permet de construire l'arborescence à partir du dendrogramme(remarquons qu'à un dendrogramme,ne correspond qu'une seule arborescence,mais l'inverse n'est pas vrai.)

SON(i) représente le 1er fils de i si ce dernier en possède au moins un.

BRO( ) représente le frère suivant de i,ou si i n'a plus de frère,BRO(i) représente le père de i,au signe près.

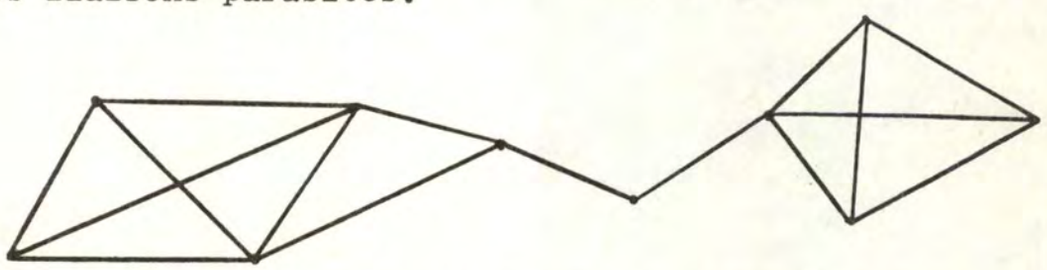
L'arborescence correspondant à la figure II.1 serait donc:



	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)
SON	-	-	-	-	-	-	-	-	-	-	1	5	11	12	9	14	13
BRO	2	-11	4	-13	6	7	-12	-14	10	-15	3	8	16	15	-16	-17	

Cette représentation sera très utile pour le retrieval,lorsqu'il s'agira de parcourir l'arborescence en vue d'obtenir les documents pertinents.

Avantages et inconvénients de la méthode SINGLE LINK.
La méthode est intéressante du fait qu'elle établit une hiérarchie dans les relations entre objets.Toutefois,comme son nom l'indique, son emploi est considérablement limité par l'existence de l'effet de chaînage:le graphe ci-dessous illustre le bruit provoqué par ces liaisons parasites.



Le même effet se produit lors de la construction des composantes connexes.
En conclusion,à un niveau h donné,les essaims formés sont complètement isolés les uns des autres(pas de recouvrement);par contre on n'a aucune mesure de la cohérence interne des essaims formés.

[JARDINE et SIBSON 68] ont tenté de supprimer le chaining effect en introduisant la notion de k-partitions(2 essaims peuvent se recouvrir par k-1 éléments au plus),mais la méthode nécessite des calculs considérables et la représentation graphique devient très complexe.

II.3.2 Les cliques maximales

La construction des cliques maximales est un processus coûteux tant au point de vue temps de calcul qu'occupation mémoire:Tous les algorithmes nécessitent en effet des accès aléatoires à la matrice associée au graphe dans lequel on recherche les cliques maximales.Cette matrice G s'obtient comme suit:étant données une valeur de cutoff fixée et la matrice de similarité S construite au moyen d'une fonction de comparaison,on a:

$S(i,j) \geq$ cutoff alors $G(i,j) = 1$

$S(i,j) <$ cutoff alors $G(i,j) = 0$

Les algorithmes connus à l'heure actuelle sont ceux de [FRANK et HARARY 62], [BONNER 64], [BIERSTONE 69], [BRON et KERBOSCH 73], [SCHNEIDER 73].Les 4 premières méthodes ont été étudiées comparativement:([BONNER 64], [AUGUSTSON et MINKER 70], [BRON et KERBOSCH 73]) de ces études,il ressort que l'on peut les classer comme suit:(par temps d'exécution croissant)[BRON et KERBOSCH,BIERSTONE,BONNER, HARARY et ROSS.
Dans la suite de ce paragraphe,nous présenterons les méthodes de BRON et KERBOSCH et une méthode(corrigée)de SCHNEIDER;puis nous exposerons l'implémentation réalisée dans cette thèse(une version modifiée de SCHNEIDER).Nous comparerons ensuite les 3 méthodes: cette comparaison ne pourra toutefois pas se faire au niveau des temps d'exécution,les langages d'implémentation utilisés étant différents.(A l'époque de nos essais,il ne nous a pas été possible d'employer un langage admettant la récursivité - Algol ou PL/I - pour l'algorithme de BRON et KERBOSCH)

II.3.2.1 La méthode de BRON et KERBOSCH

C'est une méthode de Branch and Bound,l'idée étant de pouvoir réunir au plus vite des conditions suffisantes pour affirmer l'impossibilité d'obtenir par la suite des cliques maximales.
L'algorithme consiste à appeler récursivement un opérateur d'extension manipulant les 3 ensembles suivants:

1 "compsub" contient les n° des sommets de la clique en cours de construction

2 "candidates" contient les n° des sommets susceptibles de servir à l'extension de "compsub" :tous les sommets de "candidates" sont adjacents à tous les sommets de "compsub".

3 "not" contient les n° des sommets qui ont déjà fait partie auparavant de cliques maximales et que l'on rejette pour les cliques restantes.

Les 2 derniers ensembles sont construits lors de chaque appel récursif(c'est-à-dire sont locaux à chaque appel récursif de la procédure),le premier est global et se comporte comme une pile. Initialement,"compsub" et "not" sont vides et "candidates" contient tous les n° des sommets.Puis on appelle l'opérateur d'extension défini ci-dessous:

1 Sélectionner un candidat dans "candidates"

2 Ajouter le candidat à "compsub"

3 Créer de nouveaux ensembles "candidates" et "not" à partir de ces anciens ensembles en enlevant dans ceux-ci tous les sommets non adjacents au sommet sélectionné(et l'on conserve les anciennes versions)
si "candidates" non vide:aller en 4
si "candidates" et "not" vides:imprimer "compsub";aller en 5
si "candidates" vide et "not" non vide:aller en 5

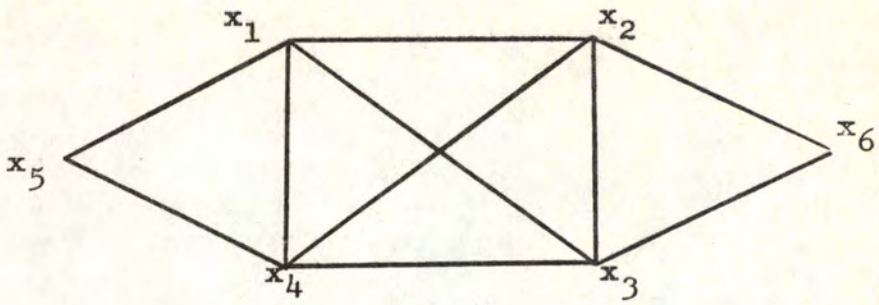
4 Appeler l'opérateur d'extension

5 (retour) -Enlever l'élément figurant au sommet de la pile "compsub" ,
-Ajouter cet élément dans "not" (et l'enlever simultanément de "candidates")
-Si "not" contient tous les sommets,alors arrêt

Une clique(identifiée dans "compsub")est maximale lorsque les ensembles "candidates" et "not" sont simultanément vides.La condition est évidente pour "candidates".L'ensemble "not" non vide signifierait que la clique trouvée est un sous-ensemble d'une clique maximale trouvée auparavant.
La méthode est du type Branch and Bound par la manière de sélectionner les candidats.En effet,si à un moment donné,un sommet de "not" est adjacent à tous les sommets de "candidates","not" sera toujours non vide et la clique non maximale.Cette condition est détectée le plus vite possible de la façon suivante:
A chaque appel de la procédure récursive,associons à tous les sommets i de "not",un compteur n_i indiquant le nombre de sommets de "candidates" auxquels il n'est pas adjacent.Lorsque nous transférons un candidat sélectionné dans "not",nous créons un nouveau compteur et les compteurs existants diminuent de 1 ou 0;mémorisons donc à chaque appel de la procédure récursive,le couple(x_i,n_i)correspondant à la plus petite valeur des compteurs et choisissons comme candidats successifs,pour étendre "compsub",des sommets non adjacents à x_i.Lorsque $n_i=0$,la condition d'arrêt est réalisée et l'on peut retourner dans la procédure de niveau inférieur.

Exemple
Nous illustrons l'emploi de l'opérateur d'extension et de la condition d'arrêt sur le graphe ci-dessous.Les cliques sont générées dans l'ordre lexicographique(le calcul du couple(x_i,n_i),lors de chaque appel récursif,ne fait que choisir les sommets dans un ordre différent,sans modifier l'emploi de l'opérateur d'extension

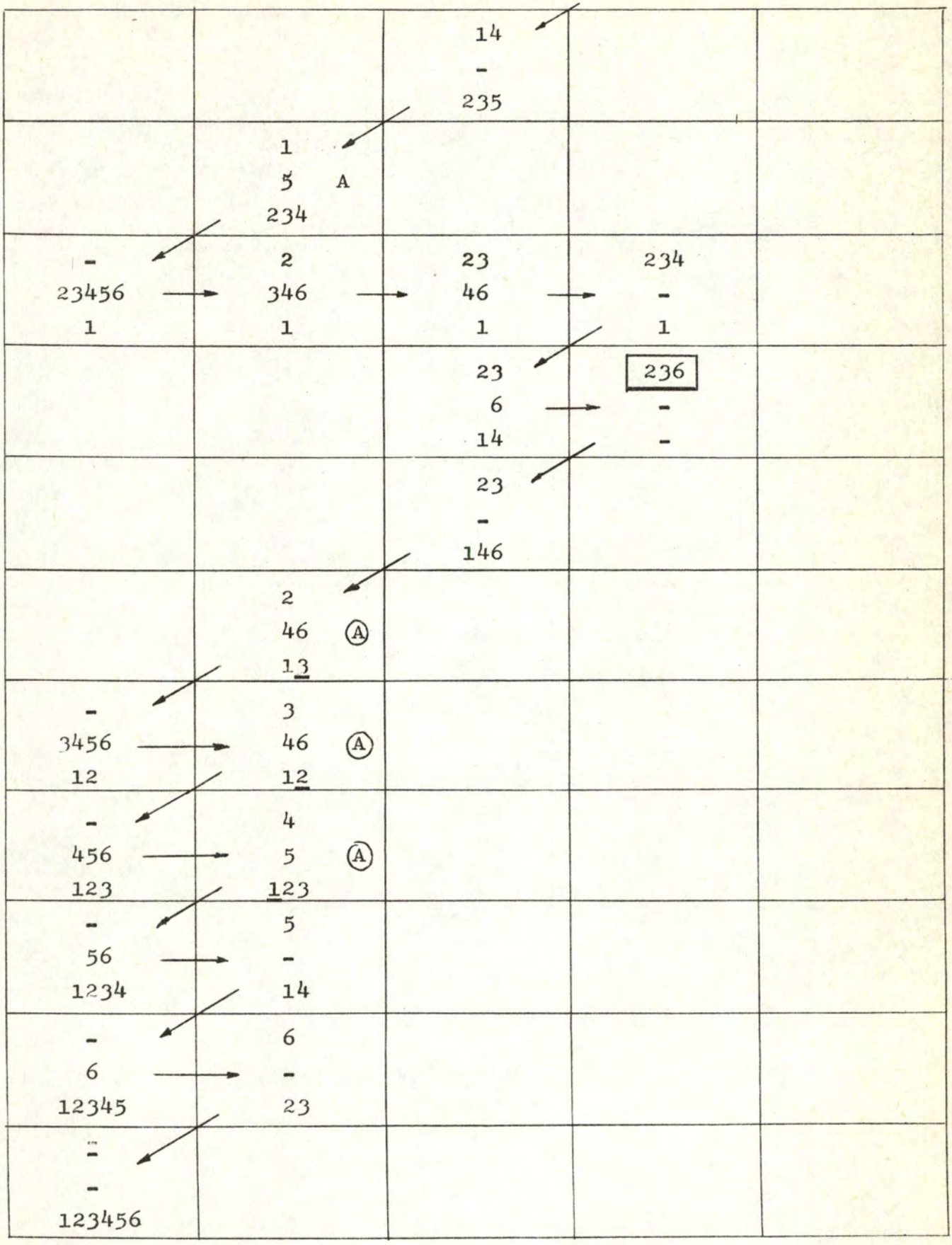
et la condition d'arrêt)



Les différentes cases du tableau illustrent l'évolution des ensembles "compsub"
"candidates"
"not"

Dans une même ligne, le passage d'une colonne à la suivante correspond à un appel récursif, c'est-à-dire à l'extension de "compsub" et à la création de nouveaux ensembles "candidates" et "not".
Le passage d'une ligne à la suivante signifie que l'on retourne dans le programme appelant: la hauteur de la pile "compsub" diminue de 1 et le sommet que l'on vient d'enlever de "compsub" modifie également les ensembles "candidates" et "not" du programme appelant.
Nous avons marqué (A) les cases du tableau pour lesquelles la condition d'arrêt est réalisée et nous avons souligné les sommets de "not" qui vérifient cette condition (rappelons qu'il suffit d'un sommet de "not" adjacent à tous les sommets de "candidates")

- 123456 -	1 2345 -	12 34 --	123 4 -	1234 = -
			123 - 4	
		12 4 (A) 3		
	1 345 2	13 4 (A) 2		
	1 45 23	14 5 23	145 - -	


14
-
235
1
5 A
234
- 2 23 234
23456 346 46 -
1 1 1 1
23 236
6 -
14 -
23
-
146
2
46 A
13
- 3
3456 46 A
12 12
- 4
456 5 A
123 123
- 5
56 -
1234 14
- 6
6 -
12345 23
-
-
123456

Discussion

La méthode s'adapte bien dans le cas où la matrice du graphe est fort creuse.Dans ce cas,en effet,la profondeur d'appel de l'opérateur de l'extension est généralement très faible.
Par contre,son efficacité diminue considérablement dans le cas de graphes presque complets(+).Dans ce cas,en effet,la profondeur d'appels est de N-1 niveaux et à chaque niveau,il faut reconstituer les ensembles "candidates" et "not" et leur réserver une zone mémoire, si bien qu'un schéma itératif semble mieux approprié qu'un schéma récursif tant au point de vue place mémoire que temps d'exécution.
La place mémoire maximale à prévoir pour le programme est de :

$N^2 + \frac{1}{2} N(N-1) + N + 2N$ mots mémoires où

N^2 correspond à la matrice du graphe.

$\frac{1}{2} N(N-1)$ correspond à l'implantation de "candidates" et"not"

N correspond à la pile "compsûb"

2N servent à mémoriser (n_i, x_i)

II.3.2.2 La méthode de SCHNEIDER

La méthode de SCHNEIDER s'explique aisément par l'emploi du formalisme suivant.
Soit un graphe $G = (X, \Gamma)$ et $V \subseteq X$.Nous dirons que P(V) est vraie si tous les sommets sont adjacents entre eux.La recherche des cliques maximales revient donc à déterminer tous les sous-ensembles P - majorants de X c'est-à-dire tous les sous-ensembles V tels que

1 P (V) soit vrai

2 P (V U $\{\alpha\}$) soit fausse

Nous présentons ici une version corrigée de l'algorithme de SCHNEIDER,l'étape 8 de la version originale pouvant provoquer l'énumération de cliques non maximales.

0 Créer une pile

1 Faire I = 1 et passer en 2

2 Si I > N on a obtenu tous les sous-ensembles P-majorants de X, sinon mettre X_I au fond de la pile;puis faire $\alpha = X_I$; $\beta = X_I$; $y = X_I$; passer en 3.

3 Placer sur la pile le premier sommet z venant après y dans l'ordre de numérotation et tel que E U $\{z\}$ vérifie P;E désignant l'ensemble des sommets contenus dans la pile.Utiliser cette procédure tant qu'il reste des sommets non encore examinés en faisant y=z.Si l'on a pu faire progresser la pile, passer en 4;sinon passer en 7.

4 Soit E l'ensemble des sommets contenus dans la pile lorsqu'on parvient à ce stade.Si $\alpha = \beta$ passer directement en 5.Dans le cas contraire,tester s'il existe un sommet γ compris entre α et β ou égal à β tel que E U $\{\gamma\}$ vérifie P.Si oui,passer en 6, sinon passer en 5.

(+) Un graphe non orienté à N sommets est dit presque complet si chaque sommet est relié à au moins N-2 autres sommets.

5 Tester s'il existe un sommet γ d'indice inférieur à I tel que E $\cup \{\gamma\}$ vérifie P .Si oui passer en 6;sinon E est une clique maximale;passer en 7.

6 Soit le sommet figurant sur la pile lorsqu'on parvient à ce stade;faire y=t;retirer ce sommet de la pile;retourner en 3.

7 Si la pile contient au moins deux éléments,passer en 8,sinon faire I=I+1;retourner en 2.

8 Soit le sommet qui figure au sommet de la pile;faire y=t;
si $\alpha = \beta$ faire β=t;enlever t du sommet de la pile;soit v le nouveau sommet;faire α=v;aller en 3.
si $\alpha \neq \beta$ faire β=t;enlever t du sommet de la pile;soit v, le nouveau sommet;faire α=min(α,v);aller en 3.

Discussion

La méthode de SCHNEIDER est un schéma itératif,générant les cliques dans un ordre lexicographique;elle nécessite N^2+N mots mémoires cette place est minimale.Par contre,le temps d'exécution est très long du fait qu'il n'y a aucune mémorisation des résultats intermédiaires;c'est le cas notamment du test P-majorant et des retours en arrière.

II.3.2.3 La version implémentée

La version implémentée dans ce mémoire a pour but de mémoriser les résultats intermédiaires les plus importants tout en conservant la simplicité de l'algorithme.Il nécessite 4 zones de mémoire auxiliaire:CANDI contient tous les sommets que l'on peut sélectionner pour une extension possible de la clique.

CANDI(J)=1 si J est un candidat possible.
CANDI(J)=0 autrement.
STACK contient les sommets de la clique.
TAKEOF contient les sommets de CANDI qui sont bloqués par la sélection d'un candidat.
MEMOR mémorise la hauteur de la pile TAKEOF après une opération d'extention(c'est-à-dire de mise à jour de STACK,CANDI et MEMOR).

Algorithme.

0 Réserver les 4 zones de mémoires STACK,CANDI,TAKEOF,MEMOR
Faire CANDI(J)=1 pour $1 \leq J \leq N$.

1 I=I+1;si $I \geq N$ stop sinon aller en 2.

2 JBEGIN=I-1; =I; =I;K=0;aller en 3.

3 Itérer cette étape jusqu'à ce qu'il n'y ait plus de candidats après STACK(K)

- Prendre le premier J>JBEGIN tel que CANDI(J)=1;K=K+1; STACK(K)=J
- Mettre à jour les 4 zones mémoires,c'est-à-dire faire CANDI(J)=0 pour tous les sommets J admissibles mais non adjacents à STACK(K);mémoriser ces sommets dans TAKEOF et mémoriser la hauteur de TAKEOF dans MEMOR
- JBEGIN=STACK(K)

4 Si l'on a pu faire progresser la pile STACK, aller en 5; sinon, aller en 7.

5 Si $\alpha \neq \beta$ tester s'il existe CANDI(J)=1 pour $\alpha < J \leq \beta$ ou $0 < J < I$
Si $\alpha = \beta$ tester s'il existe CANDI(J)=1 pour $0 < J < I$; si la réponse est oui, aller en 6; si la réponse est non, STACK contient une clique maximale, aller en 7.

6 - JBEGIN=STACK(K)
- mise à jour des zones mémoires: par consultation de TAKEOF et MEMOR faire CANDI(J)=1 pour tous les J éliminés par la sélection de STACK(K); mettre TAKEOF et MEMOR à jour
- K=K-1

7 Si K≠2 aller en 8, sinon aller en 1.

8 - JBEGIN=STACK(K)
- mise à jour des zones mémoires (cfr 6)
- Si $\alpha = \beta$ alors =STACK(K); K=K-1;
=STACK(K); aller en 3
Si $\alpha \neq \beta$ alors =STACK(K); K=K-1
=min(STACK(K), α); aller en 3

L'occupation mémoire est de $N^2 + 4N$ mots mémoires.
On peut s'attendre à ce que l'algorithme donne des performances équivalentes à la méthode de BRON-KERBOSCH dans le cas de graphes dont la matrice est très dense. L'ordre lexicographique évite la réservation en cascade de piles mémorisant des résultats intermédiaires, comme dans le cas de l'algorithme de BRON-KERBOSCH. Par contre, ce même ordre lexicographique ne permet pas d'appliquer une technique de Branch and Bound qui permettrait de rejeter au plus tôt certaines configurations.

II.3.3 Les composantes connexes.

Le graphe étant symétrique, les composantes simplement connexes se confondent avec les composantes fortement connexes. L'algorithme utilisé est celui exposé dans [FICHEFET 74].

1 Prendre un sommet arbitraire et le marquer + .

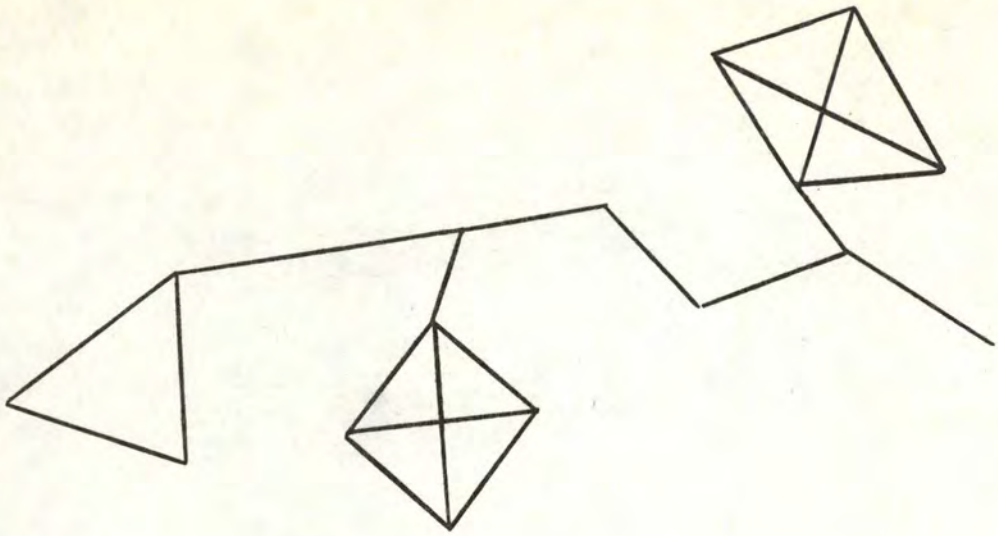
2 Marquer + tout sommet adjacent à x_i.

3 Marquer + tout sommet (non encore marqué) adjacent à un sommet déjà marqué; lorsqu'aucun sommet ne peut plus être marqué par ce procédé, les sommets marqués constituent la composante simplement connexe $C(x_i)$ contenant x_i.

4 Supprimer de G tous les sommets de $C(x_i)$.

5 Dans le sous-graphe restant, recommencer (1) à (4) jusqu'à épuisement de tous les sommets de G.

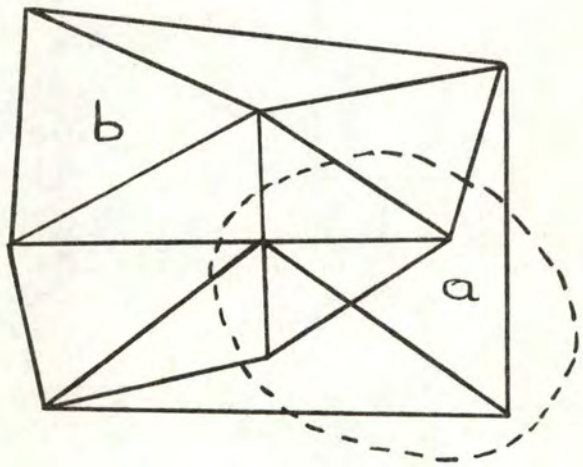
L'inconvénient principal de cette méthode est, comme dans le cas de la méthode Single-Link, l'effet de chaînage.



La méthode ne convient donc pas pour une classification automatique de termes dans laquelle on veut isoler des concepts :on ne possède en effet aucune information sur la cohérence interne des essaims.

II.3.4 Les aggrégats de NEEDHAM(clumps)

Le but de cette théorie,très ambitieuse peut-être,est de contrôler simultanément 2 paramètres technologiques de construction des essaims,à savoir la cohésion interne et le recouvrement([NEEDHAM 62],[SPARCK-JONES et JACKSON 67,70]).Dans des classifications à faible recouvrement,2 objets similaires ne peuvent figurer dans plusieurs essaims différents;en outre,chaque essaim se distingue de tous les autres par son contenu.Au contraire,un essaim est dit cohérent si chaque élément est relié à la plupart des autres éléments de cet essaim avec une corrélation supérieure à un seuil fixé.
Le principe de la méthode est de minimiser une fonction liée à ces 2 objectifs.Comme dans le cas des cliques maximales et des composantes connexes,on construit un graphe à partir d'une matrice de similarité et d'un seuil de signification.
Soit le graphe ci-dessous



L'algorithme sépare les sommets en 2 sous-ensembles disjoints a et b et évalue cette configuration au moyen d'une certaine fonction F(a,b),par exemple

$$F(a,b)= \frac{S_{ab}}{S_{aa}} \times \frac{N_a^2 - N_a}{S_{aa}}$$

où S_{ab} est le nombre de connections entre les essaims a et b

N_a est le nombre de sommets dans a.

On peut voir que la 1ère partie de cette fonction vise à former un essaim bien_isolé;la 2ème à former un essaim très_cohérent; on a en effet min S_{ab}=0(cas de composantes connexes)

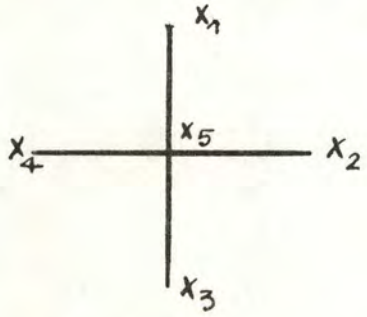
$$\max S_{aa} = \frac{N_a(N_a-1)}{2} \quad \text{(cas des cliques maximales)}$$

Une itération consiste à faire passer un sommet de b dans a(ou inversément)et à évaluer la nouvelle configuration.Un essaim est formé lorsqu'on a atteint un minimum(local) de la fonction. (voir [JACKSON 71]).
Le principe de la méthode est séduisant,car les essaims ainsi obtenus sont insensibles à l'absence accidentelle de quelques arêtes dans le graphe.L'implémentation toutefois n'est pas simple. Notons par exemple que:

- Les essaims obtenus dépendent du choix des essaims de départ.
- Certains essaims peuvent se recouvrir totalement,si bien qu'il faut une passe supplémentaire pour éliminer les essaims parasites.
- Des choix malheureux de sommets passant de b dans a peuvent rendre la convergence de l'algorithme très lente.

II.3;5 Les étoiles

Le principe est de former un essaim autour de chaque objet de la collection à essaimer:étant donné un élément de la collection, appelé centre de l'étoile,l'on recherche tous les autres éléments présentant avec le premier une corrélation supérieure à un seuil fixé(SPARCK-JONES et JACKSON 70)



Ce type d'essaim est utilisé dans les classifications de termes; comme on désire la plupart du temps des classes de taille réduite, on impose en plus une valeur maximum pour le nombre de branches de l'étoile.On remarquera que cette méthode permet de construire des relations asymétriques entre éléments d'un essaim.

II.3.6 Méthode de VASWANI

Le but poursuivi est le même que dans la théorie des clumps;mais le principe est totalement différent:on désire en effet construire les cliques maximales d'un graphe dans lequel on aurait ajouté les connections "les plus évidentes".
Si G est la matrice du graphe initial,les cliques maximales sont celles du graphe correspondant à la matrice

$G' = \lambda G + \mu G^2$ où λ et μ sont 2 scalaires dont la valeur est déterminée par expérience.

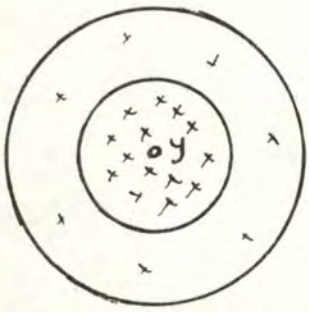
L'hypothèse fondamentale sur laquelle repose la méthode ,est que,parmi les paires de sommets non directement connectés,la proportion de paires reliés par une chaîne de longueur 2,sera beaucoup plus forte dans les essaims qui sont presque des cliques,qu' ailleurs dans le graphe.
Laméthode peut aisément être généralisée,en prenant en considération des chemins de longueur 3,4...

II.3.7 L'algorithme de ROCCHIO

Cet algorithme s'applique uniquement en recherche documentaire: son but est de limiter le nombre de documents à consulter lors d' un retrieval. L'algorithme est conçu de manière à controler simultanément le nombre,la taille et le recouvrement des essaims.Comme pour les autres méthodes d'essaimage,l'algorithme de ROCCHIO procède à partir de la matrice de similarité documents-documents; les autres paramètres d'entrée sont M_1(taille minimum d'un essaim), M_2(taille maximum d'un essaim),le nombre d'essaims,n_1,p_1,n_2,p_2 définies ci-dessous.

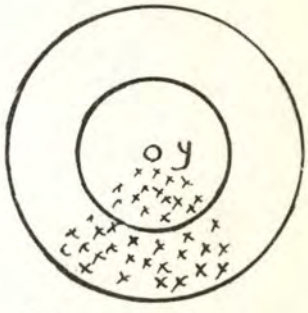
L'algorithme comporte 3 phases:

1 test de densité:ce test réalise un premier essaimage par une procédure itérative:on vérifie si un document y non encore essaimé peut-être le centre d'un essaim de documents non encore essaimés. Pour cela,il faut qu'il existe au moins n_1 documents liés à y avec une corrélation r_1(1ère condition) et n_2 documents avec une corrélation r_2(2ème condition) ($n_1 \geq n_2$ et $r_1 \leq r_2$)



1ère condition non satisfaite

2ème condition non satisfaite



1ère et 2ème conditions non satisfaites

Exemples de documents y refusés comme centre d'essaims

Si y est accepté, on ordonne les documents essaimés autour de y suivant les valeurs croissantes du coéfficient de corrélation; soit M le nombre de documents dont la corrélation avec y est supérieure à p_1. On calcule ensuite un coéfficient de corrélation p_{min} de la manière suivante:

a si $M \leq M_1$ alors $p_{min} = p_1$

b si $M > M_2$ retenir les M_2 premiers documents et passer en c.

c si $M_1 < M \leq M_2$ choisir p_{min} égal à la corrélation du document qui présente la plus grande différence de corrélation avec son suivant (rappelons que les documents sont ordonnés): cette procédure permet d'isoler un essaim du reste des documents.

2 Pour chaque essaim I, construire le vecteur de profil

$$P_I = \frac{1}{m_I} \sum_{j=1}^{m_I} \vec{d}_j \quad \text{où}$$

m_I nombre de documents dans l'essaim I

$\vec{d}_j$ vecteur représentant le jème document de l'essaim I.

Cette étape a pour but de construire le "centre de gravité" de chaque essaim. Ces vecteurs de profil peuvent être considérés comme de nouveaux centres d'essaims: on réapplique la procédure définie dans l'étape 1, chaque vecteur de profil étant cette fois comparé avec <u>toute</u> la collection.

3 Les essaims obtenus après ces 2 passes ne conviennent pas encore pour les stratégies de recherche: le recouvrement peut être trop important; le nombre d'essaims trop grand ou trop petit; une partie, souvent importante, de documents n'est pas essaimée. Il faut donc des procédures auxiliaires pour éclater ou fusionner certains essaims. Pour les documents non essaimés, diverses stratégies ont été envisagées:

- Inclure chaque document isolé dans l'essaim le plus proche
- Constituer un ou plusieurs "garbage clusters"
- Réappliquer la procédure définie ci-dessus uniquement aux documents non essaimés.

Une étude statistique des performances de l'algorithme de ROCCHIO est publiée dans (SALTON 71, chapitre XI). La collection de test consiste en 82 documents, c'est-à-dire une très petite collection. L'article ne donne malheureusement aucun détail relatif à la réalisation pratique de l'algorithme (langage d'implémentation, types de périphériques utilisés, sortie des résultats). Les temps d'exécution publiés ne semblent toutefois guère encourageants et n'incitent en aucun cas à appliquer l'algorithme de ROCCHIO à des collections comportant des milliers de documents: en effet, les temps requis pour générer les essaims de documents donnent les meilleures performances de retrieval (du point de vue recall et précision) varient entre 10 et 15 minutes et sont, de plus, très sensibles au nombre d'essaims générés.

Enfin,et ce n'est pas le moindre des inconvénients,l'algorithme implique que l'on fixe préalablement certains paramètres qui ne sont pas indépendants l'un de l'autre et qui,de plus,dépendent fortement de l'homogénéité de la collection à essaimer.

II.3.8 L'algorithme de DATTOLA

L'algorithme de DATTOLA(dont la dernière version est publiée dans [SALTON 71,chapitre XII],est le premier, et aussi le seul à notre connaissance, à présenter des temps d'exécution proportionnels à N^{α} où $1<\alpha<2$.Cet algorithme est donc conçu pour essaimer des collections comportant des milliers de documents,avec des temps d'exécution acceptables,il fonctionne selon le schéma itératif suivant

1 Construire une partition arbitraire des objets à essaimer.

2 Construire le profil de chaque essaim ainsi obtenu(c'est-à-dire création d'un descripteur de chaque essaim)

3 3.a Etant donné l'ensemble des profils,réallouer chaque objet à un ou plusieurs essaims suivant leur corrélation avec les vecteurs de profil;les objets qu'on ne peut classer dans aucun essaim sont réunis dans un nouvel essaim.

3.b Itérer la procédure(2,3.a) jusqu'à obtenir une stabilisation des essaims construits d'une itération à l'autre. La méthode n'étant pas nécessairement convergente,on imposera une limite supérieure au nombre d'itérations.

Cet algorithme présente l'inconvénient que les essaims obtenus dépendent fortement de la partition initiale,tant du point de vue nombre final d'essaims que du point de vue contenu des essaims obtenus,même si l'on cherche à contrôler la taille,le nombre et le recouvrement des essaims obtenus.Il faut donc lancer préalablement une procédure de démarrage permettant de construire les essaims pour la 1ère itération:il existe plusieurs possibilités.

a On possède à l'avance une classification plus ou moins significative des documents à essaimer:ce cas se présente notamment lorsqu'on désire réessaimer une collection de documents ayant subi des modifications importantes(ajoute et /ou suppression des documents à la collection initiale au moyen de procédures très simples.)

b On extrait un échantillon de la collection de documents,auquel on applique une procédure d'essaimage $O(N^2)$,ce qui permet d'obtenir directement les profils initiaux.

c DATTOLA propose d'autres procédures de démarrages(voir SALTON 71,chapitre XII,pp 269)de manière à obtenir des profils initiaux significatifs.

II.4 Conclusions

Nous avons exposé dans ce chapitre la plupart des techniques d'essaimage,employées à l'heure actuelle,en vue de construire des thésaurus de mots clés et de documents.

La construction d'essaims constitue un processus coûteux tant au point de vue place mémoire que temps d'exécution;les temps d'exécution sont souvent de l'ordre $O(N^2)$.Les définitions d'essaims adoptées sont généralement limitées à des définitions "technologiques",car les algorithmes définis à partir de propriétés mathématiques très précises sont impraticables à grande échelle(voir par exemple les "Ball clusters" de VAN RIJSBERGEN).

Si l'analyse par essaimage est utilisée à l'heure actuelle dans des domaines très variés,on constate que les techniques d'essaimage appliquées dans la construction de thésaurus doivent satisfaire des contraintes qu'on ne retrouve dans nul autre domaine.Par exemple,l'utilisation d'un thésaurus de mots clés implique, nous le verrons dans le chapitre suivant,la construction d'une classification non hiérarchique et à recouvrement plus ou moins arbitraire.Au contraire,les stratégies de recherche se ramènent à la consultation d'essaims obéissant à des conditions très strictes de taille,nombre et recouvrement.En passant,on remarquera combien la simplicité de la conception des méthodes d'essaimage de mots clés contraste avec les procédures très complexes utilisées pour le retrieval.La simplicité des premières méthodes n'est toutefois qu'apparente,car,(cfr chapitre suivant)des traitements préalables à l'essaimage des mots clés sont souvent nécessaires pour pondérer l'influence de certains termes;nous nous sommes limités ici à exposer des algorithmes relatifs à ces 2 types d'application.

Soulignons également que les problèmes que l'on veut résoudre par les techniques d'essaimage n'ont généralement pas de solution optimale unique.De plus,il n'existe aucun algorithme,à l'heure actuelle,dont on puisse affirmer qu'il donne une solution optimale; l'attitude la plus souvent adoptée étant d'accepter un algorithme d'essaimage s'il donne de bons résultats pour le problème envisagé,sans trop se préoccuper des propriétés mathématiques de la transformation des données.

Enfin,il est essentiel que les données à essaimer possèdent au moins la structure d'essaims,c'est-à-dire qu'il soit possible de dégager des groupes de données présentant certaines propriétés de cohérence et d'isolement;et ce n'est pas le choix d'indices de (dis)similarité ou de stratégies d'essaimage très élaborées qui permettront d'obtenir,malgré tout,des essaims significatifs. [JARDINE et SIBSON 71] définissent comme suit des données idéales pour l'essaimage: " Ideal data for cluster analysis would yield clusters so obvious that they could be picked out,at least in reasonably small-scale cases,without the need for complicated mathematical techniques and without making precise what is meant by "cluster".

En pratique,un problème d'essaimage est presque toujours caractérisé par l'absence de données idéales:Les données à essaimer se composent en effet:

- d'objets que l'on peut grouper de manière évidente.
- d'objets-ou groupes d'objets-que l'on ne peut grouper qu' avec certaines hésitations,ou après mûre réflexion,et vues différemment par plusieurs observateurs.
- d'objets pour lesquels la structure d'essaimage proposée semble manifestement inappropriée.

Le but des techniques d'essaimage(sauf pour les thésaurus de documents)est précisément d'esquisser clairement ces différentes

catégories d'objets,lorsque le nombre d'objets devient très important,et de suggérer les différents liens possibles pouvant exister entre les objets:dans ce même ordre d'idées,une matrice de (dis)similarité peut être considérée comme un résumé suffisamment complet de l'information initiale permettant de dégager les relations significatives entre objets.

CHAPITRE III CONSTRUCTION ET CONSULTATION D'UN THESAURUS

III.1 Introduction

Nous avons déjà évoqué précédemment quelques problèmes posés par la construction de thésaurus. Nous nous proposons dans ce chapitre de faire une étude microscopique de la construction et de l'emploi automatiques d'un thésaurus de mots clés, (nous employons indifféremment "mot clé", "descripteur" ou "terme" pour décrire le contenu d'un document.) en vue d'améliorer les performances d'une chaîne documentaire.
Nous supposerons que les données de départ sont constituées par un ensemble de documents, chacun étant décrit par un certain nombre de termes caractéristiques du contenu du document; nous supposerons également que la méthode d'indexation employée (manuelle ou non) n'emploie pas d'opérateurs relationnels entre termes (ces méthodes d'indexation sont discutées notamment dans [FARRADANE 74 , CROS , GARDIN , LEVY 64]).

On admet généralement qu'un thésaurus permet d'augmenter le recall d'une chaîne documentaire, le thésaurus servant d'outil de standardisation des différents langages employés par auteurs, indexeurs et utilisateurs : en d'autres termes, les thésaurus doivent permettre de comparer des documents et des requêtes décrits par des termes différents, mais traitant du même sujet.

Les deux exemples présentés ci-dessous et tirés de [SPARCK - JONES 71] montrent que l'emploi d'un thésaurus est beaucoup plus complexe qu'il n'y paraît à première vue et qu'il est souvent difficile de dissocier le recall de la précision lors d'une recherche de documents.

Exemple I

Soit une requête comprenant un seul terme a et deux thésaurus différents. Le premier ne contient que la seule classe {a,b,c, d,e} relativement à a , le deuxième contient {a,b,c} et {a,d,e}
La première classification provoquera l'énumération de tous les documents traitant de a ou b ou c ou d ou e ; la deuxième classification, bien que plus fine, donnera les mêmes résultats: nous n'avons en effet aucune raison à priori de préférer l'essaim {a,b,c} à l'essaim {a,d,e} si l'utilisateur ne précise pas davantage sa requête.
Les deux thésaurus ont essentiellement pour but d'améliorer le recall.

Exemple II

Soient les deux schémas de retrieval ci-dessous (fig.III.1 et III.2)

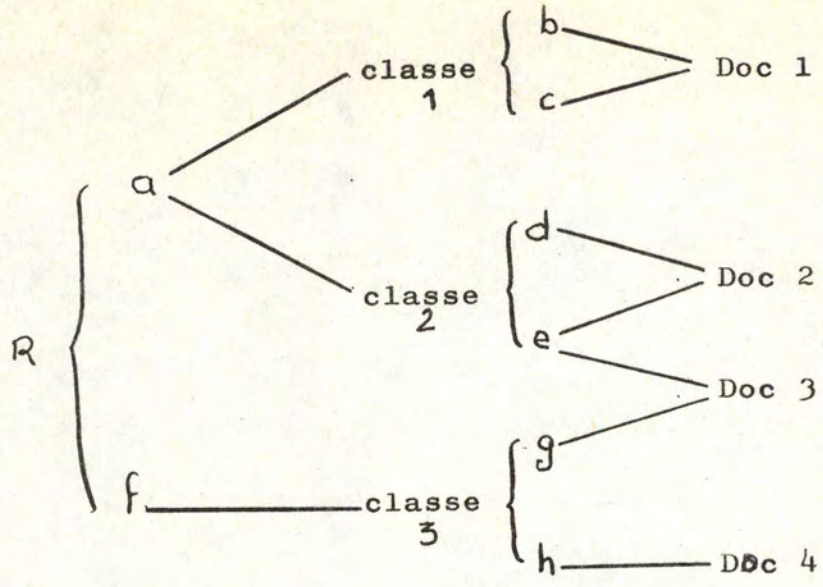


Fig. III.1

Le document D3 est de toute évidence le meilleur car il fait intervenir (indirectement) a et f
La classification améliore de nouveau le recall, puisqu'elle a pour effet de sortir le document {e,g} à partir d'une requête a,f , et ce grâce aux propriétés de substitution du thésaurus.

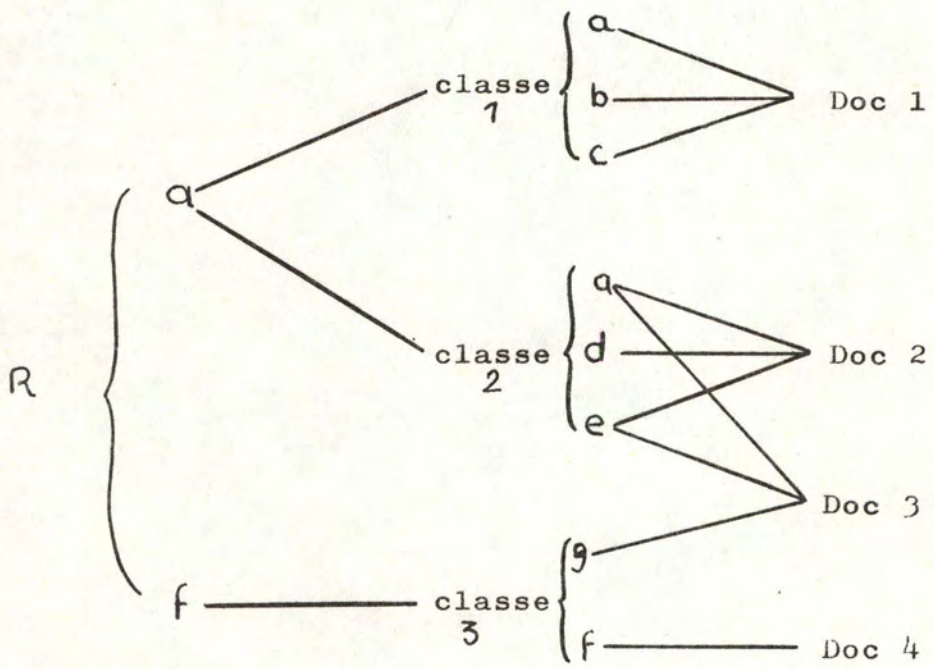


FIG III.2

Dans les deux cas, les thésaurus sont les mêmes: seules, les descriptions des documents 1,2,3 diffèrent par la présence de a .

Cette fois,la classification améliore la précision car elle sélectionne le meilleur document (n° 3) parmi ceux qui contiennent a dans leur description. Il y a néanmoins toujours un effet de recall par suite de la substitution f - g employée pour sélectionner le document n° 3 .

Ces exemples montrent combien il faut être prudent dans l'interprétation de l'effet d'une classification sur les performances de retrieval. Dans la suite de ce chapitre,nous analysons plus en détails le modèle théorique:nous examinons successivement la nature des classes du thésaurus,les contraintes que l'on souhaite imposer,leurs implications sur le matching Documents-Requêtes et les écarts à ce modèle dans une réalisation pratique.

III.2 Nature des relations entre éléments d'une classe

Le caractère mal défini des relations entre les termes d'une même classe du thésaurus a souvent donné l'impression aux expérimentateurs interprétant "de visu" les résultats obtenus, que la construction automatique d'un thésaurus introduisait beaucoup trop de bruit dans la définition des relations sémantiques entre termes:au plus,un thésaurus automatique pourrait servir de guide à une classification manuelle. Cette sensation se justifie par le fait que l'on se borne souvent à comparer le contenu sémantique des classes d'un thésaurus,sans regarder les effets d'une classification sur le retrieval des documents. Effectivement,de ce point de vue,une classification automatique n'est guère encourageante et nous allons voir pourquoi.

Dans la construction manuelle d'un thésaurus,l'indexeur constitue des familles de mots en se basant uniquement sur la signification que les mots ont par eux-mêmes,indépendamment de tout contexte: l'indexeur construit un"réseau sémantique" entre termes en faisant intervenir les relations de synonymie, quasi-synonymie,ou des relations du type générique ou spécifique. La structure du thésaurus est donc indépendante de toute description de documents appartenant à une collection particulière:ellr n'est construite qu'en fonction du bon sens.
En d'autres termes,l'indexation des documents et la construction du thesaurus sont deux processus totalement indépendants et l'on ne peut même pas garantir que les mots clés employés dans les deux processus soient les mêmes.

Au contraire,une classification automatique construit des relations de co-location entre termes:deux termes sont liés par une telle relation s'ils sont soit interchangeables,soit conjointement utilisés dans la description d'un sujet particulier. Ces relations sont établies à partir d'un relevé statistique des co-occurences de termes dans les documents.

En résumé,ces relations signifient que dans le contexte de la collection traitée,tous les termes d'une même classe peuvent être utilisés,d'une façon quelconque,pour caractériser -non pas le contenu d'un document particulier - mais une matière,un domaine de recherche,un sujet décrit dans cette collection par au moins un document.

En d'autres termes,si dans un exposé,on évoque un terme d'une classe,on peut s'attendre logiquement à ce que les autres termes de la classe y figurent également.

Comme on le voit,la construction des thésaurus automatique et manuel repose sur des hypothèses fondamentalement différentes. De plus,on voit mal comment un simple relevé statistique de co-occurences de termes pourrait établir des relations sémantiques bien précises entre termes.
Accidentellement,une classification automatique peut contenir certaines classes ne faisant intervenir que des relations particulières entre termes,par exemple de synonymie ou quasi-synonymie:mais ce n'est pas là le but d'une classification automatique.

Une étude intéressante portant sur la nature des propriétés sémantiques entre termes a été entreprise par [LESK 69] à partir d'une collection {200 documents,1179 termes} d'aéronautique. LESK conclut que suivant la valeur du coefficient de corrélation termes-termes,15 à 20 % seulement des associations entre termes sont significatives,c'est à dire peuvent être définies par des relations génériques ou de synonyme.
Par contre,lorsqu'on interprète la classification dans le contexte particulier de la collection de documents utilisée,près de 75 % des associations sont significatives: les associations ont donc une signification sémantique <u>locale</u> ,particulière à la collection étudiée.
Les classes restantes du thésaurus sont dues aux "subtilités de style"employées par chaque auteur,(par exemple,préférence d'un auteur pour certains termes). Cette expérience nous amène également à penser d'une part,qu'une classification automatique sera utilisée de façon optimale si elle opère sur les documents dont est issue cette classification et d'autre part,qu'il est intéressant qu'un même sujet soit traité dans plusieurs documents afin de bénéficier des propriétés de lissage lors du relevé statistique.

Dans ces conditions,que peut-on espérer d'une classification automatique? Il semble évident,que le modèle ne convienne absolument pas dans le cas où l'on désire établir uniquement une classification des termes,c'est à dire automatiser une classification manuelle. Par contre,si la classification a pour but d'améliorer les performances de retrieval,la classification établie peut être utile. L'on ne peut en effet affirmer,à priori, que seule une classification établie à partir de relations sémantiques bien définie,puisse être utile en retrieval:l'intérêt d'une classification réside dans son effet <u>global</u> sur le retrieval.

Le problème consiste donc à voir comment une classification, faisant intervenir un ensemble beaucoup plus vaste de relations - directes ou indirectes entre termes - non définies de manière explicite et propres à une collection particulière peut améliorer les performances de retrieval.

III.3 Conditions sur la construction des classes d'un thésaurus

Nous avons vu dans le paragraphe précédent qu'une procédure automatique ne pouvait fournir que des classes généralement mal définies du point de vue relations sémantiques, mais par contre qu'elles étaient étroitement liées à la description d'un sujet abordé dans la collection. Nous nous proposons dans ce paragraphe de déterminer des conditions permettant d'utiliser les propriétés de substitution d'un thésaurus automatique de la manière suivante:

a Le recall peut être amélioré en fournissant des synonymes ou quasi-synonymes aux termes utilisés par l'indexeur.

b La précision peut être augmentée, en ajoutant (voir introduction du chapitre, exemple II), aux termes principaux employés par l'utilisateur, et choisis en fonction de la combinaison des termes principaux (pratiquement par des techniques de pondération).

La littérature publiée jusqu'à ce jour, abonde en techniques d'essaimage, évaluant les thésaurus obtenus par un certain nombre de paramètres mesurables, à savoir:

1 Cohérence interne ou densité des liaisons entre les termes d'une classe.

2 Le caractère symétrique ou non des relations dans une classe.

3 Degré de recouvrement des classes.

4 La proportion des mots communs et leur distribution dans les classes du thésaurus.

5 Nombre de classes.

6 Taille moyenne des classes et distribution de la taille des classes.

7 Distribution de la fréquence de consultation des termes à l'intérieur d'un essaim.

8 Distribution de la fréquence de consultation des différents essaims.

Ces paramètres se limitent malheureusement à caractériser les aspects techniques relatifs à la construction et la consultation d'un thésaurus. Nous préférons dans ce paragraphe, formuler les conditions sur la structure et le contenu des classes.
Les techniques d'essaimage que nous pouvons utiliser se répartissent grosso modo en quatre grandes classes(fig.III.3)

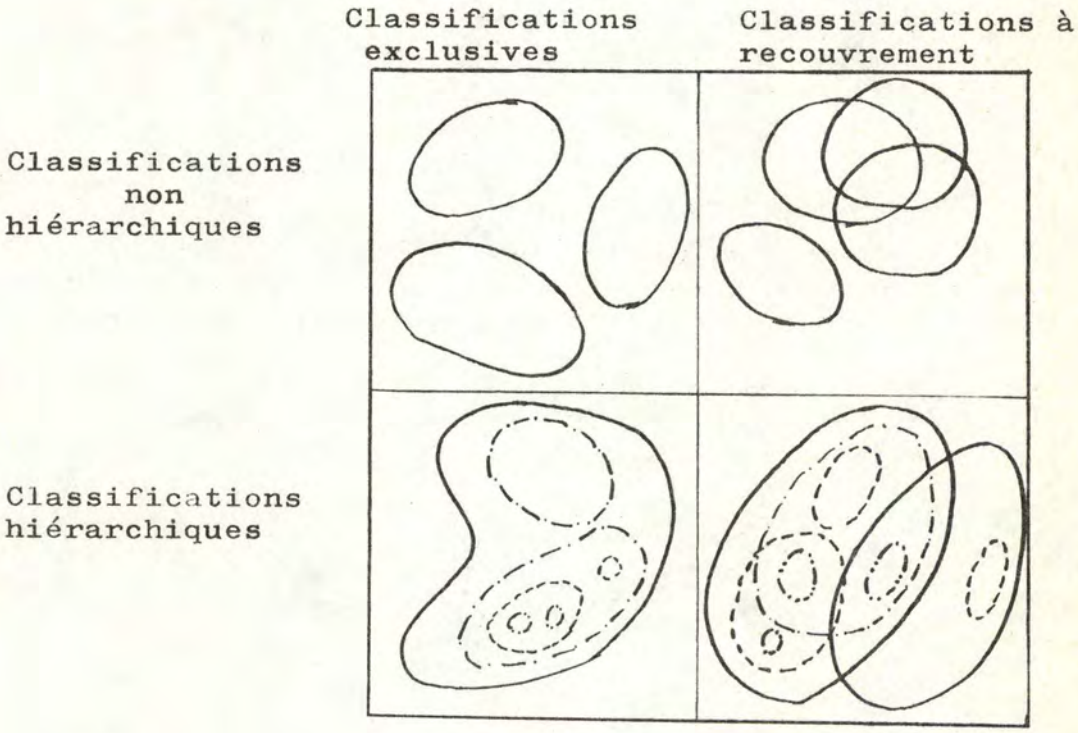


Fig III.3 Classifications des techniques d'essaimage.

Les classifications que nous désirons sont à recouvrement, car l'appartenance d'un mot à plusieurs classes permet de dégager les différentes significations de ce mot dans la collection, et donc de préciser davantage le contenu d'un document ou la formulation d'une requête.
Les classifications hiérarchiques sont séduisantes, car elles simulent un réseau sémantique de relations entre termes. Nous avons vu, toutefois, dans le paragraphe précédent, que la nature des informations recueillies, ne permettait pas de construire pareil réseau, de manière satisfaisante.
De plus, ces relations construites à partir des combinaisons des termes utilisés pour décrire les documents seraient beaucoup trop confuses, incomplètes et même contradictoires, vu le caractère superficiel et parfois aléatoire de la description des docu-

ments;ces relations ne seraient d'aucune utilité pour le retrieval. Même les classifications hiérarchiques sont insuffisantes pour construire de pareilles relations.
Considérons par exemple,le graphe des relations suivant(fig.III.4)

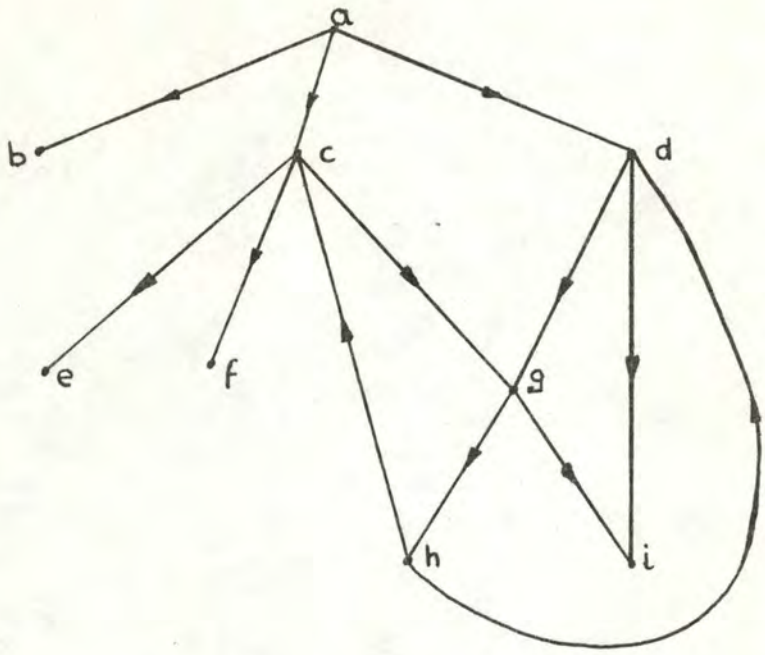


Fig. III.4

Un algorithme,s'il existe,permettant d'expliciter pareille structure serait totalement inefficace et de plus inutile pour le but poursuivi,car les techniques d'essaimage visent précisément à n'extraire des données initiales,que des informations significatives et suggestives:cet algorithme ne conviendrait manifestement pas pour ce type de données. Enfin,une structure hiérarchique est trop rigide et trop contraignante dans l'optique que nous nous sommes fixés,car les rapports entre deux termes donnés,peuvent,suivant l'usage qu'en font les auteurs,tantôt être de nature synonymique,tantôt de nature générique.

En résumé,nous demanderons aux classes d'un thésaurus d'obéir aux principes suivants:

1 Les classes ne contiennent que des termes ayant une forte tendance à survenir simultanément dans les mêmes documents(on désire éviter des groupements accidentels de mots).

2 Chaque classe doit évoquer un concept,une idée;dans cette optique,les classes doivent pouvoir se recouvrir librement pour permettre l'appartenance d'un terme à plusieurs classes suivant les différentes significations de ce terme.

3 L'appartenance d'un mot à une classe répond à la loi du tout ou rien:on n'évalue pas le degré d'appartenance d'un mot à une classe(ce qui serait possible dans le cas

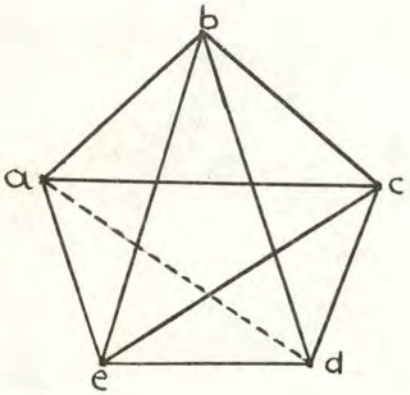
de classifications hiérarchiques).

4 Les classes formées doivent être stables:

- Deux descriptions légèrement différentes des mêmes documents doivent donner des thésaurus pratiquement identiques.
- Les classes du thésaurus ne doivent pas être perturbées par de petites erreurs dans la description des documents.
- L'ajoute ou la suppression de quelques documents ne doit pas bouleverser une classification établie.

5 Un terme n'a pas de position privilégiée dans une classe:tous les termes d'une même classe peuvent être substitués l'un à l'autre lors de l'emploi du thésaurus dans le retrieval.

Parmi les classifications non hiérarchiques et à recouvrement,celles qui ont donné les résultats expérimentaux les plus intéressants du point de vue retrieval sont les cliques maximales,les étoiles et les aggrégats de NEEDHAM.
Les composantes connexes et les chaînes de mots ne conviennent pas par suite du "chaining effect" et peuvent provoquer la présence de plusieurs cncepts distincts dans une même classe.
D'un point de vue théorique,les aggrégats devraient normalement donner les meilleurs résultats:les cliques maximales imposent une cohérence interne totale de l'essaim et aucune contrainte sur l'environnement de cet essaim;les aggrégats ont été conçus pour être moins sensibles que les cliques à des oublis de certaines arêtes dans le graphe des relations termes-termes,oublis dus aux aléas de la description des documents.
Par exemple,si nous considérons le graphe suivant(fig.III.5)



L'absence de l'arête ad provoquera la génération de 2 cliques maximales a,b,c,e et b,c,d,e et d'un seul aggrégat a,b,c,d,e

Fig. III.5

L'oubli de l'arête ad est manifestement accidentel;mais on ne peut en déduire automatiquement que le retrieval sera meilleur en utilisant des aggrégats plutôt que des cliques maximales:les trois exemples suivants montrent des situations beaucoup plus difficiles à trancher:(fig.III.6)

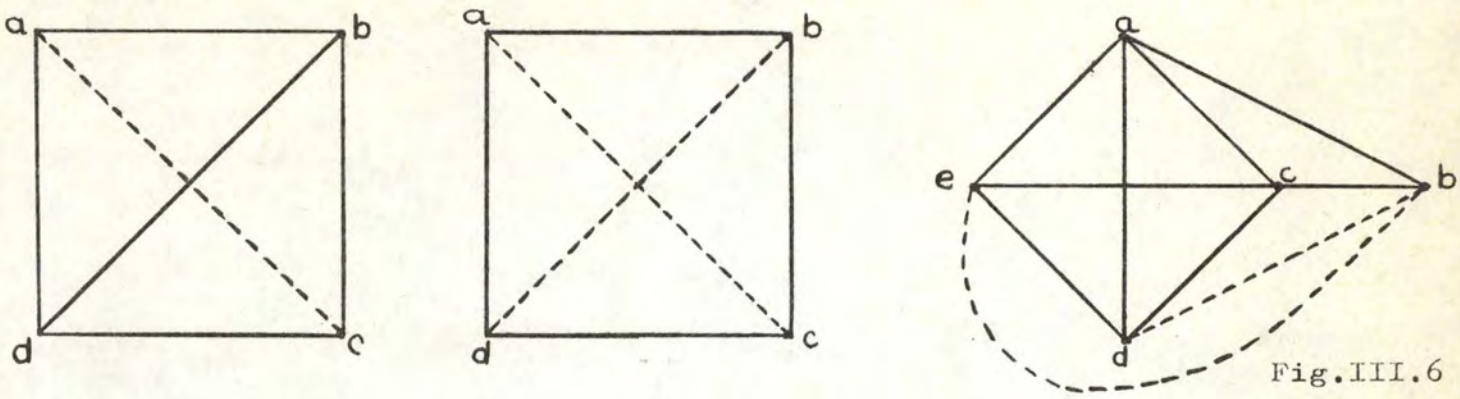


Fig.III.6

Peut-on considérer que l'absence des arêtes ac dans le 1er cas, ac et bd dans le 2 ème cas, bd et bc dans le 3 ème cas, soit réellement due à une description défectueuse des documents concernés? On pourrait tout aussi bien estimer, par exemple qu'il existe plusieurs concepts(abd et bcd pour le 1er cas, acde et abc pour le 3ème cas) ou aucun concept bien défini(2ème cas).

Il en va de même pour la comparaison aggrégat-étoile; ces dernières étudient les relations entre termes autour d'un terme central bien fixé: [SPARCK-JONES 71] cite des cas pathologiques: ces situations proviennent du fait que le terme cible ne figure pas nécessairement au centre d'un essaim. Par exemple, (fig.III.7)

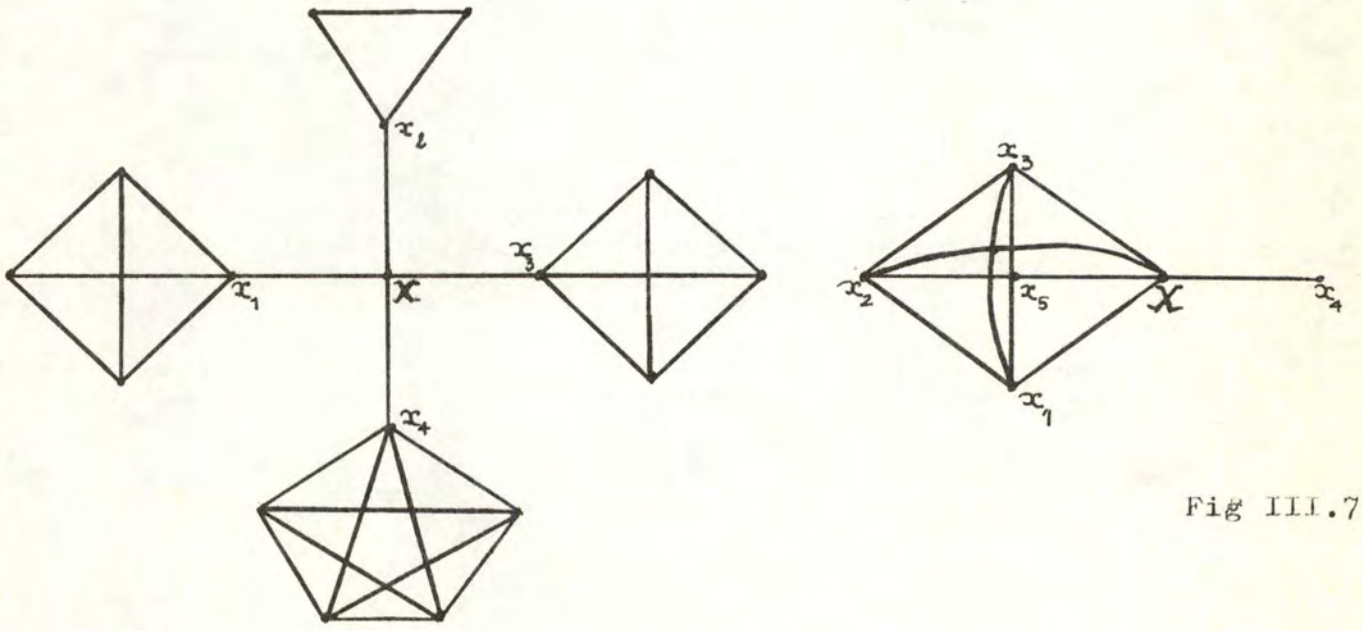


Fig III.7

Les étoiles diffèrent des autres essaims par deux points

-La construction des étoiles est très rapide.

-Les relations entre les termes d'une étoile peuvent être orientés.

Nous ne développerons pas davantage de considérations théoriques sur le choix des techniques d'essaimage. Des études récentes effectuées par [SALTON 72], [SPARCK-JONES 69], [SPARCK-JONES et JACKSON 70], [SPARCK-JONES et BARBER 71], indiquent que les trois méthodes -aggrégats, cliques maximales, étoiles- donnent des résultats équivalents du point de vue retrieval, lorsqu'elles sont utilisées de façon optimale (voir les conditions de définition des classes).

Remarque.
Accessoirement, on peut également s'intéresser à une classification des termes pour la comparer à une classification manuelle. On peut combiner plusieurs techniques d'essaimage afin de dégager des informations relatives à la nature et la densité des relations termes-termes. Au cours de nos expériences par exemple, nous avons utilisé simultanément la méthode Single-Link, les cliques maximales et les composantes connexes, et fait varier la valeur de cutoff pour la construction de la matrice de similarité.

III.4 Utilisation en retrieval d'un thésaurus construit automatiquement

Notre but, dans ce paragraphe, est de préciser comment l'on peut utiliser les propriétés de substitution des termes d'une classe d'un thésaurus afin d'augmenter le recall et/ou la précision d'une chaîne documentaire.
Intuitivement, on conçoit qu'un thésaurus puisse être utilisé de deux façons différentes:

a On remplace les termes par les classes contenant ces termes (fig.III.8).

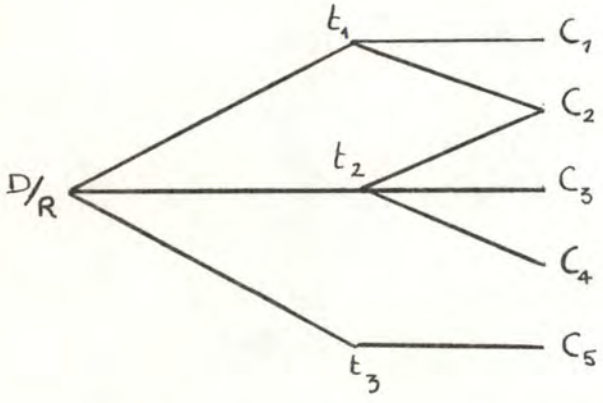


Fig III.8

Le document ou la requête est décrit par(t1,t2,t3) ou par(c1,c2,c3,c4,c5)

b On remplace les termes d'une requête ou d'un document par les termes des classes contenant les termes originaux (fig.III.9)

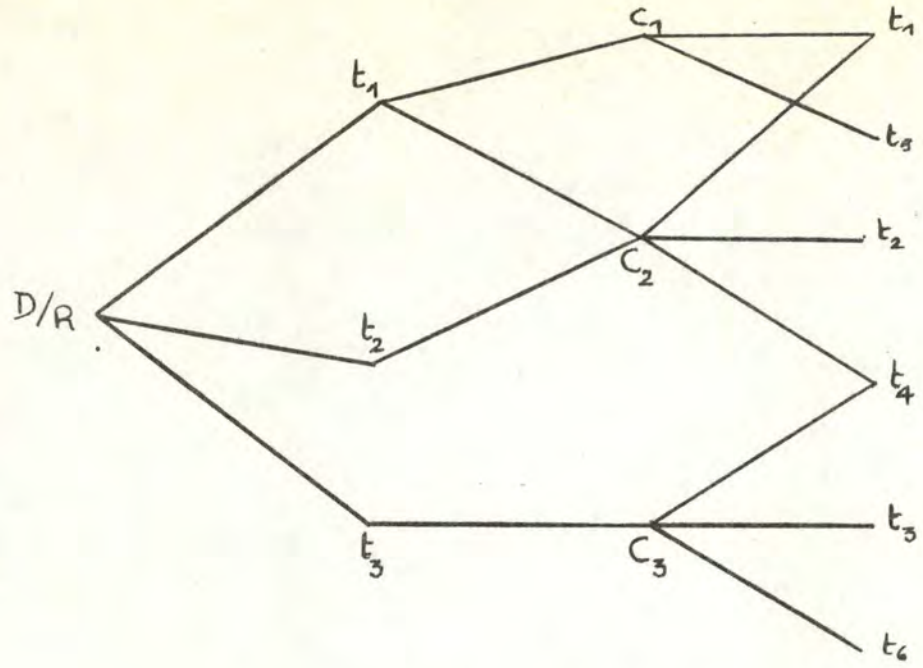


Fig III.9

On passe de la description(t1,t2,t3)à la description (t1,...,t6)

Avant d'exploiter ces transformations, nous analyserons d'abord la signification et les conséquences des transformations effectuées par l'emploi du thésaurus.

Nous avons vu dans le paragraphe III.2,que les thésaurus automatiques étaient construits à partir de l'usage des termes dans la collection particulière étudiée et non à partir de la signification que les termes ont par eux-mêmes;et c'est le relevé des co-occurences terme-terme qui fixe l'usage de ces termes.

En conséquence,transformer une description initiale selon l'un des deux schémas ci-dessus,revient en fait à modifier les propriétés statistiques des mots et donc à altérer leur usage initial (cfr. [SPARCK-JONES et BARBER 71]).

Les conséquences de cette transformation doivent être envisagées sous deux aspects.

1 Complétude de la description d'une requête ou d'un document.

2 Spécificité des mots utilisés comme vocabulaire de description.

Prenons par exemple la deuxième transformation:son effet est double:D'une part,un document est décrit par davantage de termes, ce qui rend sa description plus complète qu'initialement.

D'autre part,nous voyons que le vocabulaire initial n'a ni augmenté,ni diminué,mais que les mots de vocabulaire sont beaucoup plus souvent employés.

Comme les termes sont jugés d'après leur usage,ils ont acquis une signification plus générale,c'est-à-diresont devenus moins spécifiques.A la limite,un mot,qui après application d'une des transformations,figurerait dans la description de tous les documents,serait devenu totalement inutile.

On voit mal l'intérêt d'appliquer de telles transformations aux documents,car elle pourrait altérer plus ou moins profondément,l'image du contenu que l'indexation a voulu conférer aux documents.
Par contre,elles peuvent s'appliquer aux requêtes,sans diminuer la spécificité des termes,car l'utilisateur est bien souvent incapable de définir de manière précise et complète sa question et peut même ne pas avoir une idée claire de la signification des termes qu'il emploie.
Remarquons également que si nous employons la première transformation,nous devons formuler et les requêtes et les contenus de documents avec les classes,alors que la seconde transformation permet un nombre bien plus grand de possibilités.

Explicitons la première transformation par l'exemple suivant(fig.III.10)

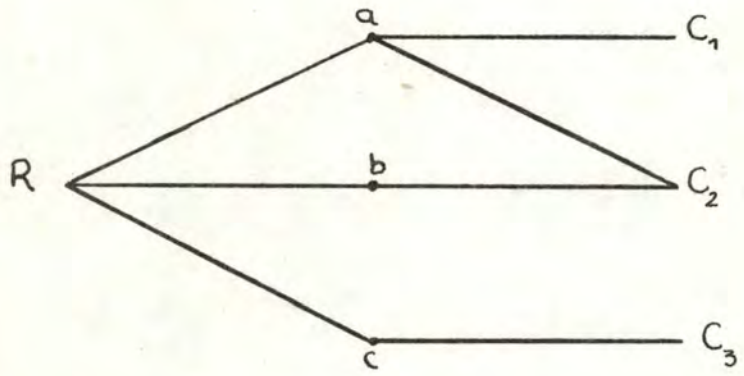


Fig III.10

Il existe de nombreuses possibilités d'utilisation des classes du thésaurus:

1 (c1,c2,c3)Recherche des documents dont la description en classe se rapproche le plus de la requête(c1,c2,c3)

2 (c1)(c2)(c3) On décompose la requête en trois sous-requêtes: on désire avoir un recall important.

3 (c2) On recherche les documents traitant de c2 uniquement,car c2 est le seul à contenir plus d'un terme de la requête.

4 (c1/1,c2/2,c3/1) La notation est identique à celle employée dans 1 sauf que c1 a un poids 1,c2 un poids 2,c3 un poids 1.
Cette technique est la plus intéressante,car elle exploite les classes du thésaurus de façon naturelle:

- -La pondération ne repose sur aucun critère arbitraire,mais se déduit logiquement de la structure du thésaurus.
- -Les deux significations de a(que l'on déduit de son appartenance à c1 et c2) sont pondérées

différemment;et de plus,c'est la combinaison (a,b)des termes employés par l'utilisateur qui permet de préciser le sens de a et de pondérer en conséquence:ceci est dû aux propriétés de recouvrement des classes.

Nous préférons toutefois la deuxième transformation,car on désire souvent garder souvenance,lors du matching Document-termes, des termes de la requête initiale.
Soit schéma suivant:(fig.III.11)

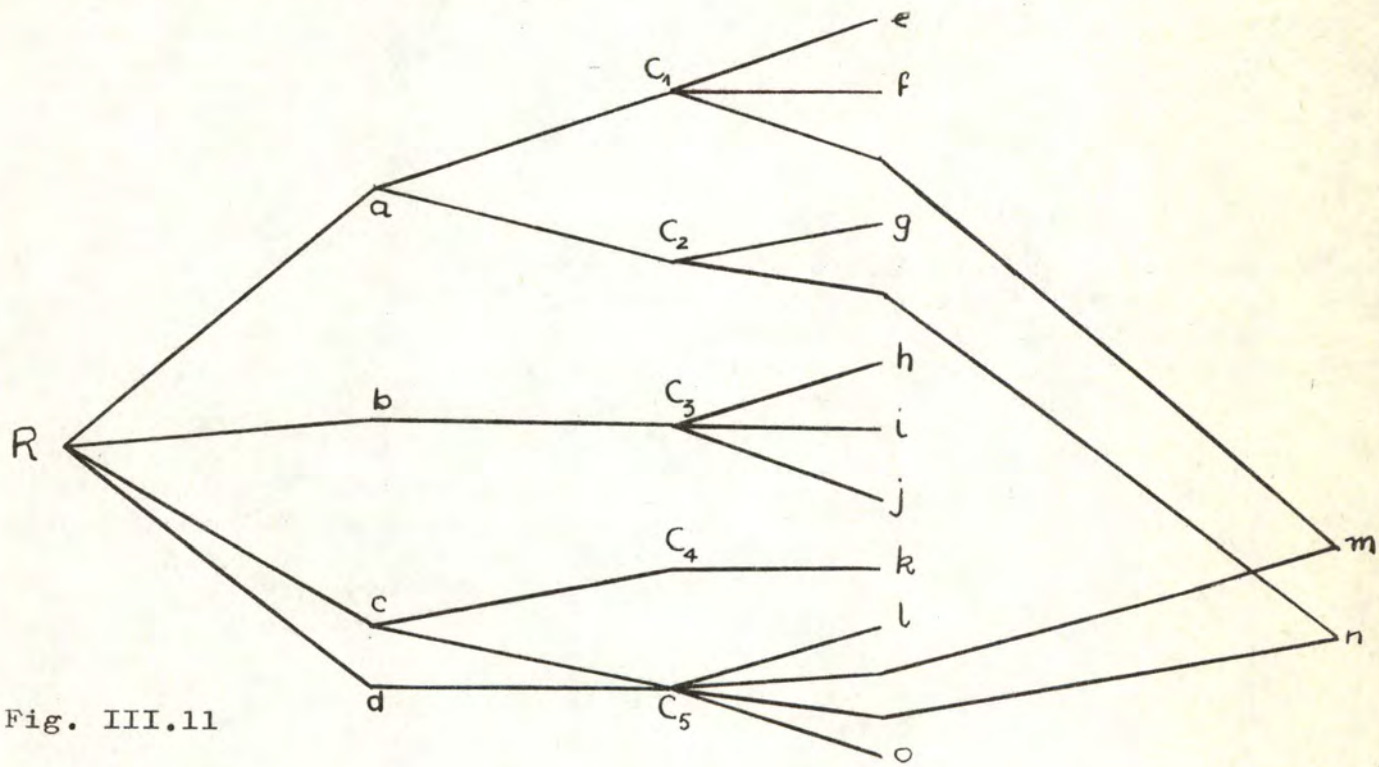


Fig. III.11

Le matching Document-requête peut également s'opérer de très nombreuses façons:

1 (a,b,...o)

2 (a/2,b/2,c/2,d/2,e/1,...o/1)
Le matching tient compte des termes de la requête initiale en leur accordant un poids plus important.

3 (c,d,l,m,n,o):On ne reprend que les termes de la classe c5,car elle est la seule à contenir plus d'un terme de la requête.

4 (a,b,c,d,l,m,n,o)

5 (a/2,b/2,c/2,d/2,m/1,n/1):Cette méthode utilise beaucoup plus finement les possibilités du thésaurus: alors que les quatre premières prenaient la requête initiale plus tous les termes d'une ou plusieurs classes,celle-ci en sélectionne certains appartenant à certaines classes.Dans le cas traité,par exemple,

m et n figurant dans plus d'une classe,on peut considérer qu'ils sont étroitement liés à la requête initiale et estimer qu'ils sont les seuls intéressants pour la requête concernée.

6 (a/3,b/3,c/3,d/3,m/2,n/2,e/1,...1/1,o/1)

7 (a/3,b/3,c/3,d/3,m/3,n/3,e/1,...,1/1,o/1)

8 Une autre catégorie de choix consisterait à choisir tous les termes d'une classe,pourvu que cette classe contiennent au moins deux termes de la requête initiale:ici,les termes ajoutés seraient e,f,g,k,l,m,n,o et de pondérer de manière automatique leur influence.

Nous voyons qu'il existe une multitude de façons d'étendre la requête initiale au moyen du thésaurus,selon le but poursuivi. La manière de procéder est toujours la même:

-Matching direct sur les termes originaux.

-Matching indirect sur les termes ajoutés par consultation du thésaurus,la sélection et le poids de ces termes étant fonction de l'importance relative accordée au recall et à la précision;la méthode présente l'avantage que la pondération n'est pas arbitraire.

III.5 Problèmes liés à la fréquence d'emploi des termes

Le schéma de construction et de consultation que nous avons explicité repose sur l'hypothèse fondamentale que la distribution de fréquence d'emploi des termes du vocabulaire est uniforme.En pratique,cette hypothèse n'est jamais vérifiée et son effet sur les performances de retrieval peut être catastrophique.Rappelons en effet,que les termes fréquents n'ont pas beaucoup de signification,de par leur nature même,alors que les termes rares sont très spécifiques.
Pratiquement,un terme fréquent figure dans de nombreuses classes du thésaurus;comme,de,plus,les requêtes sont elles aussi,le plus souvent,exprimées avec des termes fréquents;il en résulte que le matching documents-requête s'opère surtout sur des termes sans signification,ou sur des substituts induits par ces termes, et non sur des termes spécifiques du vocabulaire,alors que le but des thésaurus est précisément de proposer des alternatives pour les termes rares qui auraient été "oubliés" dans une requête.
Il faut donc limiter les possibilités de substitution des termes fréquents et promouvoir celles relatives aux termes rares.
La stratégie la plus souvent utilisée est celle-ci:

1 On repère tous les termes fréquents(par exemple,tous les termes intervenant dans plus de 10% de la description des documents)et on les extrait du vocabulaire initial.

2 On constitue 2 thésaurus:l'un avec les termes fréquents en construisant des classes exclusives;l'autre avec les termes restants,le seuil de signification étant choisi de manière à permettre un recouvrement assez important.

Cette stratégie permet de limiter l'influence des termes fréquents,tout en ne les éliminant pas(ils sont en effet nécessaires pour extraire du fonds documentaire les ouvrages à intérêt général.

III.6 Stratégies de recherche

Les thésaurus de documents ont pour but de réduire le travail de recherche en n'examinant qu'un petit nombre de documents susceptibles de répondre à une requête.Les stratégies de recherche reposent sur l'idée suivante:si l'on a groupé tous les documents en un nombre réduit d'essaims,chaque essaim ne contenant que des documents traitant du même sujet,le processus de recherche consisterait simplement à repérer les seuls essaims susceptibles de contenir les documents pertinents,puis de balayer séquentiellement tous les documents de ces essaims.Autrement dit, il existe une association(statistique)étroite entre tous les documents répondant à une même question(cluster hypothesis).Il existe 2 grandes stratégies de recherche:

1 Méthode de SALTON

Elle suppose que les documents ont été préalablement essaimés par l'algorithme de ROCCHIO;la procédure de recherche se fait en 2 étapes;

a Comparaison de la question avec tous les vecteurs de profil("centroïd vectors")permet de choisir les essaims à consulter.

b Consultation séquentielle des documents des différents essaims retenus et classement des documents sélectionnés suivant la corrélation document-question

Les vecteurs de profil peuvent également être essaimés:on obtient de cette façon un "Directory" à 2 niveaux.
Le schéma suivant illustre le procédé de recherche.

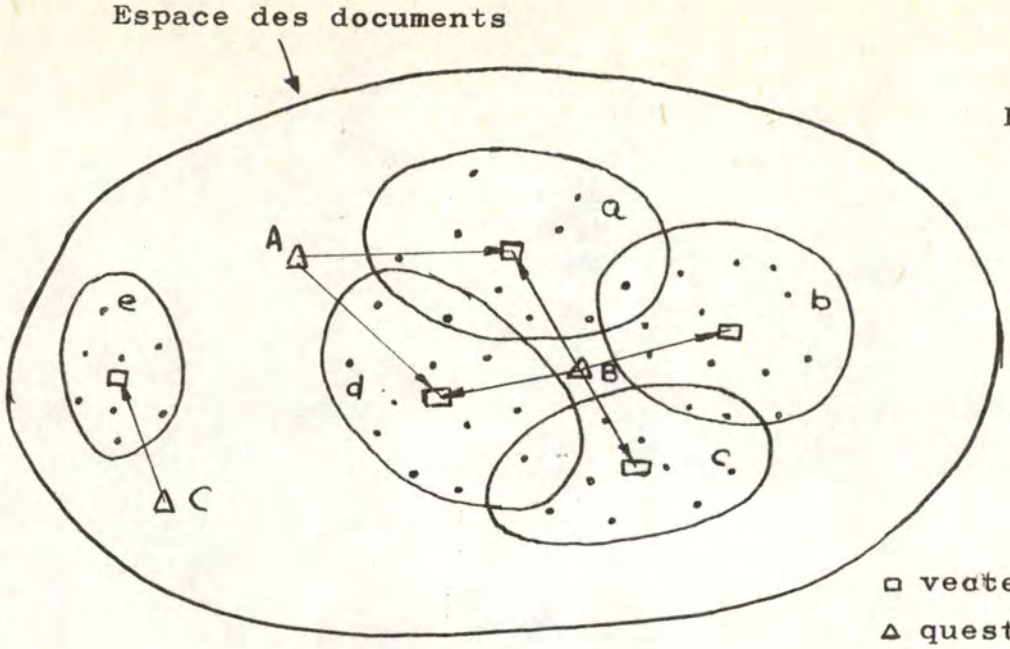


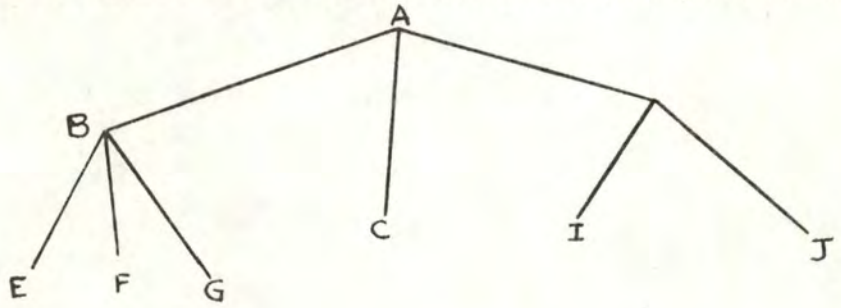
FIG III.12

a,b,c,d,e,les essaims
A,B,C, les questions

La question A amène à rechercher les documents parmi a et d, la question B parmi a,b,c et d,la question C parmi e.
Les performances de retrieval de cette procédure de recherche dépendent fortement des conditions techniques imposées à l'algorithme de ROCCHIO(pas d'essaim réduit à un seul document,recouvrement très faible).
L'algorithme semble malheureusement très sensible au nombre d'essaims alors que le retrieval est optimal lorsque le thésaurus contient de nombreux essaims.

2 Méthode de JARDINE et VAN RIJSBERGEN

JARDINE et VAN RIJSBERGEN proposent un retrieval basé sur une classification hiérarchique des documents,connue sous le nom de "Broad Search Strategy".Les noeuds terminaux de l'arborescence représentent les documents.A chaque noeud intermédiaire A,correspond un vecteur de profil,représentatif de l'ensemble des noeuds terminaux du sous-graphe partiel de racine A.
La figure ci-dessous illustre une partie de l'arborescence



Supposons que l'on doive étudier le noeud A;on étudie son 1er fils,soit B:3 cas peuvent se présenter:

a Le profil B ne ressemble en rien à la question;on abandonne B et tous ses descendants et on passe à l'étude de C.

b Le profil de B ressemble de manière "imprécise"à la requête:on examine successivement les descendants de B,puis on passe à l'étude de C.

c Le profil de B correspond "suffisamment"à la requête on considère comme pertinents tous les documents correspondant à la racine B,on passe ensuite à l'étude du noeud C.

Le parcours de l'arbre se fait à l'aide de 2 fonctions f (Q,N)et g(Q,N)et 2 paramètres x et y fournis par l'utilisateur;Q et N sont les représentations de la question et du profil du noeud N

Si $f(Q,N) \leq x$ alors arrêt de l'étude du noeud N

Si $f(Q,N) \geq x$ et $g(Q,N) \geq y$ alors énumération de tous les noeuds terminaux correspondant à la racine N

Si $f(Q,N) \geq x$ et $g(Q,N) \leq y$ alors étudier les fils de N

(Un exemple de fonctions f et g est présenté au chapitre suivant)

Le choix de la valeur des paramètres x et y permet d'orienter le parcours de l'arborescence,au gré de l'utilisateur.

Comparaison avec l'algorithme de SALTON:

a Contrairement à la méthode de SALTON,on est assuré qu'un document donné ne sera comparé au plus qu'une seule fois avec la requête.

b Les documents isolés doivent également recevoir un traitement particulier,sinon ils sont directement liés à la racine du graphe.

c La méthode peut également perdre de son efficacité par suite des effets de saturation des vecteurs de profil correspondant aux noeuds proches de la racine:Les associations entre groupes de documents sont en effet de moins en moins fortes au fur et à mesure que l'on remonte vers la racine;cet effet est encore accentué par le "Chaining effect".

d La méthode peut être étendue avec quelque succès à des collections assez importantes de documents(une technique semblable pourrait d'ailleurs être implementée dans la méthode de SALTON):on part d'un noyau de documents que l'on essaime;on espère que la structure ainsi conférée au thésaurus sera proche de celle qui correspondait au thésaurus de toute la collection; puis on introduit successivement les documents restants que l'on traite comme des questions et on les attache aux noeuds présentant la plus forte corrélation.

III.7 Conclusion

Nous avons voulu dans ce chapitre,d'une part, énoncer des conditions nécessaires pour obtenir une bonne classification,exploitable pour le retrieval,et d'autre part,expliciter les mécanismes permettant de formuler,à partir d'une question initiale,une nouvelle question,plus complète et mieux adaptée au vocabulaire employé pour l'indexation des documents: Le but de cette transformation est de séparer davantage les documents répondant et ne répondant pas à une question.

Le fait que les classes d'un thésaurus automatique sont souvent mal définies ne doit pas nous amener à conclure qu'un thésaurus est moins intéressant qu'un thésaurus manuel.Simplement,la définition des classes est différente:dans un thésaurus manuel,les relations entre termes sont déterminées d'après leur signification; alors que dans un thésaurus automatique,elles sont établies d'après leur usage et il est normal que ces relations ne soient pas toujours clairement définies,étant donné que les descriptions des documents comportent elles-mêmes une part d'arbitraire.
Une classification automatique doit être considérée comme un élément d'une chaîne documentaire introduit pour en améliorer les performances.Les critères à respecter sont:

- Les termes d'une classe doivent être fortement corrélés entre eux.
- Il faut limiter les possibilités de substitution des termes fréquents et encourager au contraire celles des termes non fréquents.
- Le couplage Documents-termes est effectué en pondérant de manière différente les termes intervenant dans le couplage; cette pondération est automatique et déterminée en fonction de l'importance relative accordée au recall et à la précision.

Il n'existe pas de procédure simple d'essaimage de documents, car on doit modifier la structure "naturelle" du thésaurus pour satisfaire les contraintes techniques d'efficacité.
Nous n'avons pas abordé les problèmes liés à la dynamique d'une collection de documents(voir [BRAVEN 70] [LESSER 69] [JOHNSON et LA FUENTE 70], [SALTON 72]).On peut les étudier sous 2 aspects différents:

1. Ajouter,retrancher,modifier petit à petit les documents sans trop altérer la structure existante du thésaurus,ou au contraire,construire des procédures itératives de réessaimage,les modifications apportées étant devenues trop importantes.
2. Modifier progressivement la description du document,afin de promouvoir les documents les plus souvent demandés et de pénaliser les documents qui ne sont presque jamais utilisés.

CHAPITRE IV.

SYSTEME EXPERIMENTAL.

IV.1 Description générale

Ce chapitre présente le système expérimental implémenté dans ce mémoire.Comme Data Base,nous avons choisi un ensemble de livres de la bibliothèque de l'Institut d'Informatique.Les livres couvrent les domaines suivants:anlyse numérique,programmation mathématique, théorie des graphes,probabilités et statistique ,processus stochastiques,simulation,théorie des langages,Operating Systems.
La collection consiste en environ 300 documents indexés par 550 mots clés.Les livres ont étés répertoriés en 3 grandes catégories correspondant plus ou moins à celles établies dans la revue"Computing Reviews",des ACM .

Les associations entre mots clés et entre documents ont été calculées au moyen de la corrélation sinusoïdale.Les mots clés ont été essaimés à partir des cliques maximales,des composantes connexes et de la méthode du Single-Link(classification hiérarchique exclusive).Cette dernière méthode a été également appliquée pour construire l'arborescence relative aux documents.Le thésaurus est constitué à partir des cliques maximales.Le seuil de signification choisi pour construire le graphe est situé dans la zone de stabilité relative aux cliques maximales,c'est-à-dire dans la zone où le nombre et la taille de ces cliques sont très peu sensibles à de petites fluctuations de ce seuil.Nous n'avons fait aucun traitement relatif à la fréquence des termes utilisés dans l'indexation.
Dans la construction de l'arborescence,nous avons limité à 20 le nombre de valeurs possibles du coefficient de dissimilitude,afin de limiter le nombre de niveaux et de pouvoir tracer le deudrogramme correspondant sur une imprimante d'ordinateur.
La consultation automatique du thésaurus permet de compléter ces questions et de pondérer les termes initiaux et les termes ajoutés au gré de l'utilisateur.Ce dernier fournit également les valeurs des paramètres de sévérité selon l'importance relative accordée au recall et à la précision.L'arborescence peut aussi être parcourue sélectivement,ce qui évite un balayage séquentiel de toute la collection de documents.
L'ensemble de l'application a été implémenté sur un ordinateur SIEMENS 4004-151 à mémoire virtuelle(capacité 1000 K).Le langage utilisé est en grande partie le FORTRAN IV G.;le compilateur FORTRAN utilisé demande toutefois quelques dispositions particulières pour les E/S sur disque(il n'est pas nécessaire de définir des DEFINEFILE ou des variables associées,chaque enregistrement non formatté nécessite 16 bytes supplémentaires gérés par le système).
Le package présenté permet de traiter un maximum de 1000 documents indexés par un maximum de 1024 mots clés.Les limitations sont imposées par le stockage de la matrice du graphe.La construction et l'adressage de cette matrice ont été réalisées en ASSEMBLER/360 afin de réduire l'occupation mémoire à 128 K.

IV.2 Methode d'indexation employée.

La méthode d'indexation employée est une méthode superficielle et manuelle.Plusieurs personnes peuvent indexer les documents de chacune des grandes catégories;mais pratiquement,la plupart d' entre elles sont traitées par une même personne.Un livre est indexé à son entrée en bibliothèque.L'indexation est basée d'une part sur une consultation rapide et superficielle du document par l'indexeur concerné,par exemple à partir de la table des matières, de l'introduction,du résumé,ou d'un survol très rapide du contenu de l'ouvrage et d'autre part sur le "background" de l'indexeur dans ce domaine.L'indexeur décrit le document par un ensemble de mots clés.Chaque mot clé est constitué par un terme -ou groupe de termes- jugés significatifs du contenu des documents.Les termes employés par l'indexeur peuvent éventuellement être différents de ceux employés par l'auteur du document.Il n'y a pas de discipline particulière imposée aux indexeurs.De même,il n'y a pas de consultation préalable d'un dictionnaire des mots clés déja employés afin d'assurer la cohérence et la complétude de l'indexation: bref,il n'y a pas de contrôle du vocabulaire employé.Une certaine cohérence est donc assurée par le fait que les livres d'une même catégorie sont,la plupart du temps,indexés par une seule et même personne.
La description d'un document n'est plus modifiée par la suite. Seules,quelques corrections mineures ont été apportées lors de l'acquisition des documents afin de faciliter le codage.Par exemple,un groupe de termes a été éclaté en plusieurs mots clés,traduisant ainsi plusieurs idées,ou encore,on n'a conservé que la version française d'un même concept figurant tantôt en français, tantôt en anglais.Cette méthode peut être considérée comme l'une des méthodes manuelles les plus simples,les plus simples et demandant le moins d'efforts aux indexeurs.Les aléas de cette méthode d'indexation peuvent être illustrés par l'exemple suivant rencontré dans notre implémentation:Les documents ci-dessous ne contiennent pas le mot Operating Systems dans leur description:

COFFMAN et DENNING: Operating System Theory

COHEN: Operating System analysis and design.

BARRON: Computer Operating Systems.

Par contre,ce mot clé figure dans:

CLAPP: A description of the internal operation of the ADAM System.

JACKSON: The design and implementation of communications languages for Digital Computers.

Manifestement,l'indexeur a estimé que le mot Operating System était superflu dans la description des documents de la 1ère catégorie.Par contre,il a voulu souligner le rôle non évident des Operating Systems dans les sujets abordés dans la 2ème catégorie. Signalons enfin que les documents de la collection étudiée,de par leur nature même,sont d'intérêt assez général:il faudrait s'attendre à une proportion importante de termes fréquents;ces problèmes

d'indexation sont abordés plus en détail dans les chapitres V et VI.Le codage consiste à associer un n° di à chaque document et un n° tj à chaque mot clé.La description du jème document se fera de la manière suivante:

(dj / ti,tj,tk,tℓ,...tp)

Les données d'entrée se présentent donc sous forme d'une matrice rectangulaire:

d_1	t_4	t_2	t_8	t_6				
d_2	t_3	t_5						
d_3	t_1							
d_4	t_4	t_7	t_{40}					

Les données sont liées par le module RWDATA qui lit séquentiellement la matrice des données ligne pour ligne.
Une carte se présente sous la forme dj t1 t5 ... tp suivant le format FORTRAN(I5,25I3). Si la description d'un document comporte plus de 25 termes,une 2ème carte peut être utilisée juste derrière la 1ère,et où figure à nouveau le n° du document concerné.
La fin de lecture de cette matrice est détectée par la présence d'un n° négatif de document.
Le module RWDATA transforme cette matrice en une autre matrice rectangulaire T où T(i,j)=p=poids du mot clé tj dans le document di. Dans notre cas,puisqu'il n'y a pas de traitement des termes selon leur fréquence,on a tij=1 ou 0.Cette matrice est envoyé sur disque ligne par ligne sur le fichier n° NFICH5 et colonne par colonne sur le fichier n° NFICH1.

IV.3 Choix d'un coéfficient de similitude.

Nous avons choisi la corrélation cosinusoïdale comme coéfficient de similitude:ce coefficient est valable quel que soit le type de données -binaires ou non- et il est employé avec succès dans de nombreux problèmes d'essaimage.

Si X_{ti} désigne le cème vecteur colonne de T,on obtiendra pour valeur du coefficient de dissimilarité des termes ti et tj

$$S(ti,tj)=\cos(\vec{X}_{ti},\vec{X}_{tj}) = \frac{\vec{X}_{ti}*\vec{X}_{tj}}{\|\vec{X}_{ti}\|\|\vec{X}_{tj}\|}$$

La matrice est envoyée après multiplication par un facteur constant(10.000 dans la routine COSINE)sur le fichier n° NFICH3.(En constituant une matrice de nombres entiers,on peut stocker en INTEGER*2 au lieu de REAL*4)
Ces opérations sont effectuées dans la 1ère partie de la routine COSINE.

La corrélation cosinusoïdale présente l'inconvénient d'exiger de nombreuses opérations de multiplication ;on lui préfère parfois la mesure de TANIMOTO:

$$S(t_i, t_j) = \frac{\alpha_{ij}}{\alpha_{ii} + \alpha_{jj} - \alpha_{ij}}$$

où α_{ij} = nombre de documents décrits simultanément par les mots clés ti et tj.

La mesure de TANIMOTO est plus rapide à calculer:cette dernière considération peut être prépondérante dans le choix d'une méthode lorsqu'on sait qu'il y a $\frac{N_t(N_t-1)}{2} + \frac{N_D(N_D-1)}{2}$ coefficients à calculer(N_t = nombre de mots clés et N_D = nombre de documents.)

La mesure de TANIMOTO s'applique toutefois uniquement à des données binaires.

De même,si $\vec{Y}_{d_i}$ représente la ième ligne de T,la similitude entre les documents d_i et d_j s'exprime par:

$$S(d_i,d_j) = \cos(\vec{Y}_{di}, \vec{Y}_{dj})$$

Dans la construction du dendrogramme,nous travaillerons plutôt avec le coéfficient de dissimilitude,multiplié par 20 et ramené à des valeurs entières(voir routine COSINE)

$$DC(d_i,d_j) = 20 - \text{int}((S(d_i,d_j)\ 20)+0.5)$$

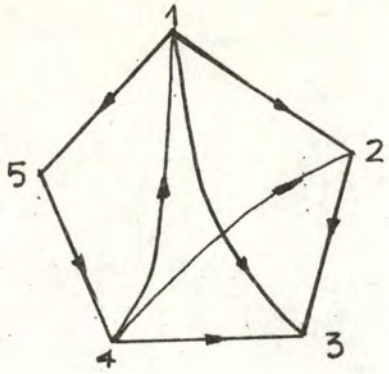
Le dendrogramme correspondant possèdera donc au maximum 20 niveaux.

IV.5 Construction des cliques maximales et des composantes connexes

Les algorithmes ont été présentés dans le chapitre II. Nous donnons ici quelques détails complémentaires relatifs à l'implémentation de ces routines.Une zone mémoire a été réservée au moyen d'un COMMON étiqueté ZONAUX afin d'éviter la transmission de trop d'arguments.

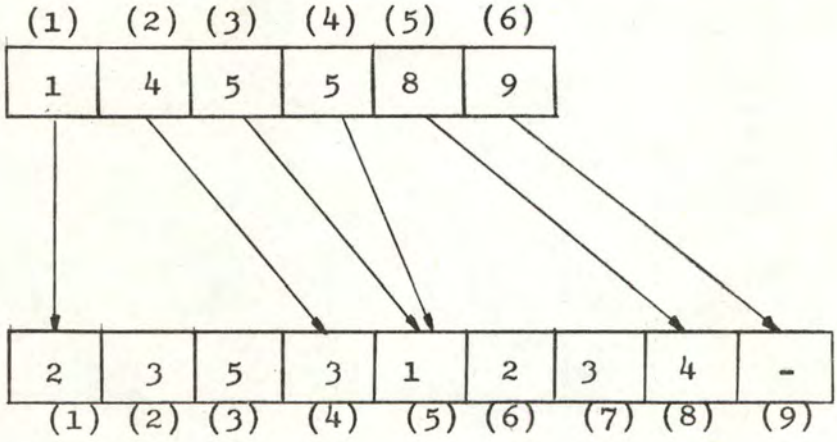
La matrice du graphe correspondant à un seuil de signification donné, est stockée telle quelle,ligne par ligne,un élément de la matrice n'occupant qu'un seul bit.On réduit de cette façon la place mémoire d'un facteur 8(128 K autotal).La consultation de la matrice au moyen de la routine BITEST est d'autre part très rapide.Cette routine comporte en effet(outre le sauvetage et la restauration des registres)le positionnement sur le byte contenant 6 bits à consulter,au moyen de la formule (I-1)*128+J-1 pour l'élément(I,J) de la matrice du graphe,et un test sous masque. Nous avons choisi cette méthode plutôt que celle qui consiste à mémoriser le graphe par dictionnaire,de la manière suivante:

Soit un graphe G.



1	2	3	5
2	3		
3			
4	1	2	3
5	4		

Son implémentation pourrait se faire au moyen de 2 vecteurs.



En se basant sur une densité de 48%,on aurait une occupation mémoire de 80 K bytes(les cases du dictionnaire prennent chacune 2 bytes).
La consultation d'un dictionnaire n'est pas très performante:il faut en effet,en moyenne plusieurs comparaisons et plusieurs calculs d'adresses(même en travaillant par recherche dichotomique).
La 1ère méthode a de plus l'avantage de nécessiter une place fixe en mémoire,indépendante de la nature des documents et du seuil de signification.
Une routine ENUMER permet de traiter les résultats obtenus:par un jeu de switches dynamiques,on peut demander d'imprimer le thésaurus des documents et / ou des mots clés et de mémoriser ces thésaurus sur une périphérique.

IV.5 Construction du dendrogramme

La méthode peut s'appliquer indifféremment aux mots clés et aux documents,pourvu que l'on ait la matrice des DC correspondants.

La routine SLINK est le noyau de la méthode:

a Elle lit la matrice triangulaire inférieure des DC, ligne par ligne ou appelle un programme (COSLNK pour l'essaimage des documents). Après la lecture de chaque ligne, elle appelle SLINK 1 qui met à jour la représentation par pointeurs(

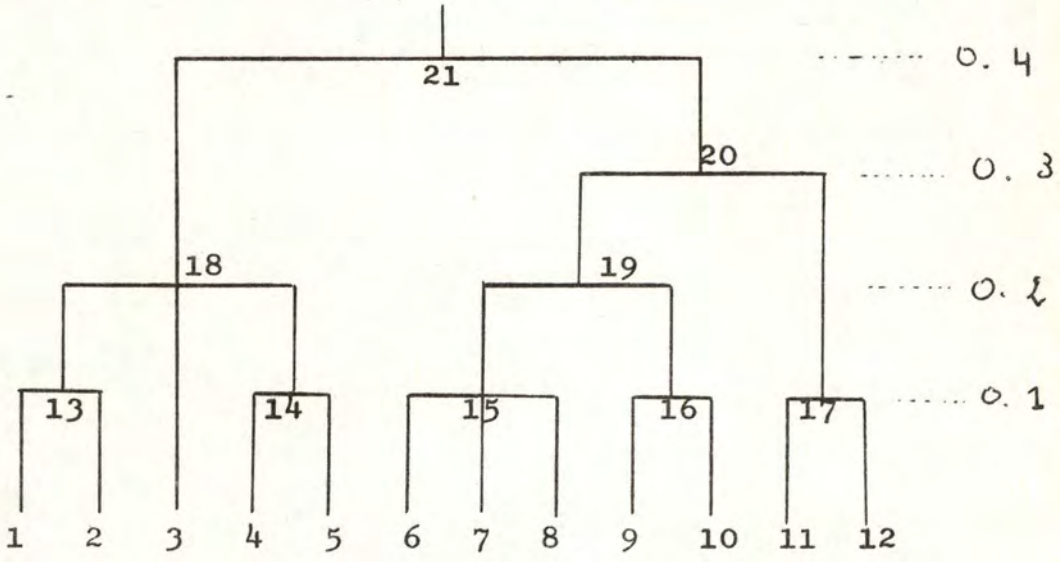
b Elle appelle SLINK2 afin de transformer la représentation par pointeurs en une représentation permettant de tracer le deudrogramme.

c Elle appelle la routine DENDR pour l'impression du dendrogramme; un switch permet d'orienter l'impression selon que les objets à essaimer sont des mots clés ou des documents.

d Elle appelle la fonction SLINK3 qui calcule la distorsion imposée par l'ultramétrique.

IV.6 Construction de l'arborescence

La représentation(Π, λ)des pointeurs est mémorisée dans les vecteurs(PISOM,LAMBDA). On préfère toutefois travailler avec une représentation où LAMBDA construit les valeurs des niveaux, rangées par ordre croissant. Ceci nécessite un vecteur supplémentaire NUM qui mémorise les anciennes positions dans LAMBDA. Les figures ci-dessous montrent la méthode utilisée.



	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)
λ	0.1	0.2	0.2	0.1	0.4	0.1	0.1	0.2	0.1	0.3	0.1	∞
Π	2	5	5	5	12	8	8	10	10	12	12	12

Le fonctionnement de l'algorithme est expliqué par l'évolution des 4 zones ci-dessous:

	1	2	3	4	5	6	7	8	9	10	11	12
NUM	1	4	6	7	9	11	~~7~~ 13	3	~~8~~ 15	~~10~~ ~~16~~ 19	~~5~~ ~~14~~ 18	~~12~~ ~~17~~ 20
PISOM	2	5	8	8	10	12	~~8~~ 14	~~8~~ 14	~~10~~ 16	~~12~~ 17	~~12~~ ~~17~~ 20	~~12~~ 20
LAMBDA	0.1	0.1	0.1	0.1	0.1	0.1	0.2	0.2	0.2	0.3	0.4	
NEWNAM	1	~~2~~ 13	3	4	~~5~~ ~~14~~ 18	6	7	~~8~~ 15	9	~~10~~ ~~16~~ 19	11	~~12~~ ~~17~~ 20

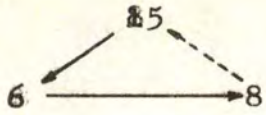
	1	2	3	4	5	6	7	8	9	10	11	12
SON	0	0	0	0	0	0	0	0	0	0	0	0
BRO	2	-13	14	5	-14	7	8	-15	10	-16	12	-17

	13	14	15	16	17	18	19	20	21			
SON	1	3	6	9	11	13	15	19	18			
BRO	3	-18	16	-19	-20	20	17	-21				

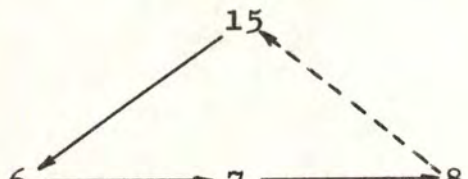
L'algorithme comporte autant d'itérations qu'il y a de niveaux différents dans le dendrogramme.L'algorithme traite d'abord les noeuds terminaux,puis les noeuds de niveau immédiatement supérieur(ceux-ci peuvent être alors considérés comme de nouveaux noeuds terminaux).Cette manière garantit que lorsqu'on crèe,un noeud, tous les fils de ce noeud existent déjà.
Un autre vecteur NEWNAM est nécessaire pour mémoriser les"nouvelles dénominations" des noeuds.Par exemple,on voit qu'il existe des relations(2,5) et (3,5) au niveau h=0,2.Toutefois,lorsque les noeuds 13 et 14 ont étés créés,ces relations deviennent(13,14) et(3,14).
A chaque itération,la procédure effectue les opérations suivantes

1 Mise à jour des nouvelles dénominations des pointeurs.

2 a Pour la création d'un nouveau noeud:on étend BRO et SON d'une case et on les met à jour.Par exemple,supposons qu'on rencontre(en parcourant NUM et PISOM)la relation (6,8).La 1ère case libre étant la 15ème,on crée SON(15)=6;BRO(6)=8;BRO(8)=-15.



- Pour l'insertion d'un élément dans une chaîne existante: le nouvel élément se met toujours en 2ème position dans la chaîne. Par exemple, supposons qu'on rencontre (7,8)



-BRO(8) → 15
SON(15) → 6
BRO(6) → 7
BRO(7) → 8
} Insertion de 7 entre 6 et 8

Les vecteurs BRO et SON sont ensuite mémorisés sur le fichier NFICH9.

Une 2ème phase dans la construction de l'arborescence consiste à former les vecteurs de profil des noeuds intermédiaires, c'est à dire à résumer le mieux possible le contenu des documents ayant ce noeud comme racine. Il existe une bonne part d'arbitraire dans le choix des méthodes admissibles: nous avons décidé de prendre la somme vectorielle des vecteurs représentant les noeuds terminaux concernés. Accessoirement, nous ajoutons aussi le nombre de ces noeuds terminaux concernés.

IV.7 Consultation de l'arborescence

Une question est dépouillée dans EXQUES et étendue par consultation du thésaurus. On obtient ainsi une question Q dont les composantes valent 0,1,2;3... selon les poids accordés. La routine BSTRA(Broad Search Strategy) parcourt l'arborescence et décide (selon l'algorithme exposé dans le chapitre II) s'il faut descendre ou non dans l'arborescence. Les 2 quantités à calculer sont F qui détermine si le noeud est acceptable ou non et G qui détermine, pour un noeud accepté, si l'essaim pris globalement répond suffisamment bien à la question.

Soient REP la représentation vectorielle d'un noeud N correspondant à un essaim C.

Q la question

C le nombre de noeuds terminaux figurant dans C.

Q le nombre de termes dans Q.

T_Q l'ensemble des indices J tels que $Q(J) > 0$.

TE (Total Evidence) = $\sum_{J \in T_Q} REP(J)$.

On peut alors construire $F = \frac{TE}{Q}$ et $G = \frac{TE}{C}$.

Ces quantités sont calculées dans FGCQ.
Si le couplage essaim-question atteint la précision souhaitée, la routine LISDOC est appelée afin d'énumérer tous les documents figurant dans l'essaim retenu.

CHAPITRE V

RESULTATS EXPERIMENTAUX

Dans ce chapitre,nous exposons et nous interprétons les résultats obtenus à partir des trois collections suivantes:

Collection I:Documents d'analyse numérique,algèbre,analyse fonctionnelle,théorie des graphes

Collection II:Documents de théorie des probabilités,statistique,processus stochastique et simulation

Collection III:Documents de théorie des langages,operating systems et fichiers.

V.1 Caractéristiques générales de la collection de documents

Le tableau ci-dessous constitue un bref résumé des caractéristiques des trois collections

	I	II	III
nombre de documents	108	63	129
nombre de termes	194	120	213
nombre de requêtes	16	17	15
nombre moyen de termes par document	7.1	3.6	4.4
nombre moyen de documents par terme	3.47	1.88	2.68
nombre de termes très rares(1 fois)	41%	60.7%	54%
nombre de termes rares(1 et 2 fois)	53.5%	79.4%	71%
nombre de termes intervenant dans plus de 10% de la collection	6.2%	5%	1.88%
nombre de documents décrits par un seul terme	5.5%	27%	3.88%
nombre de documents décrits par un ou deux termes	12%	44.5%	26.7%
nombre moyen de termes employés par -requête initiale	2.31	1.23	2.6
-requête complétée par thésaurus	4.87	4.4	5.55
nombre moyen de documents énumérés dans les cas suivants:			
-requête initiale et description initiale des documents	7.95	2.53	7.46
-requête initiale et description des doc. complétée par thésaurus	8.62	2.53	-
-requête complètée par thésaurus et description initiale	8.18	3.78	6.94

-requête complétée par thésaurus et description compl. par thés.	8.56	4.4	-

Tableau V.1

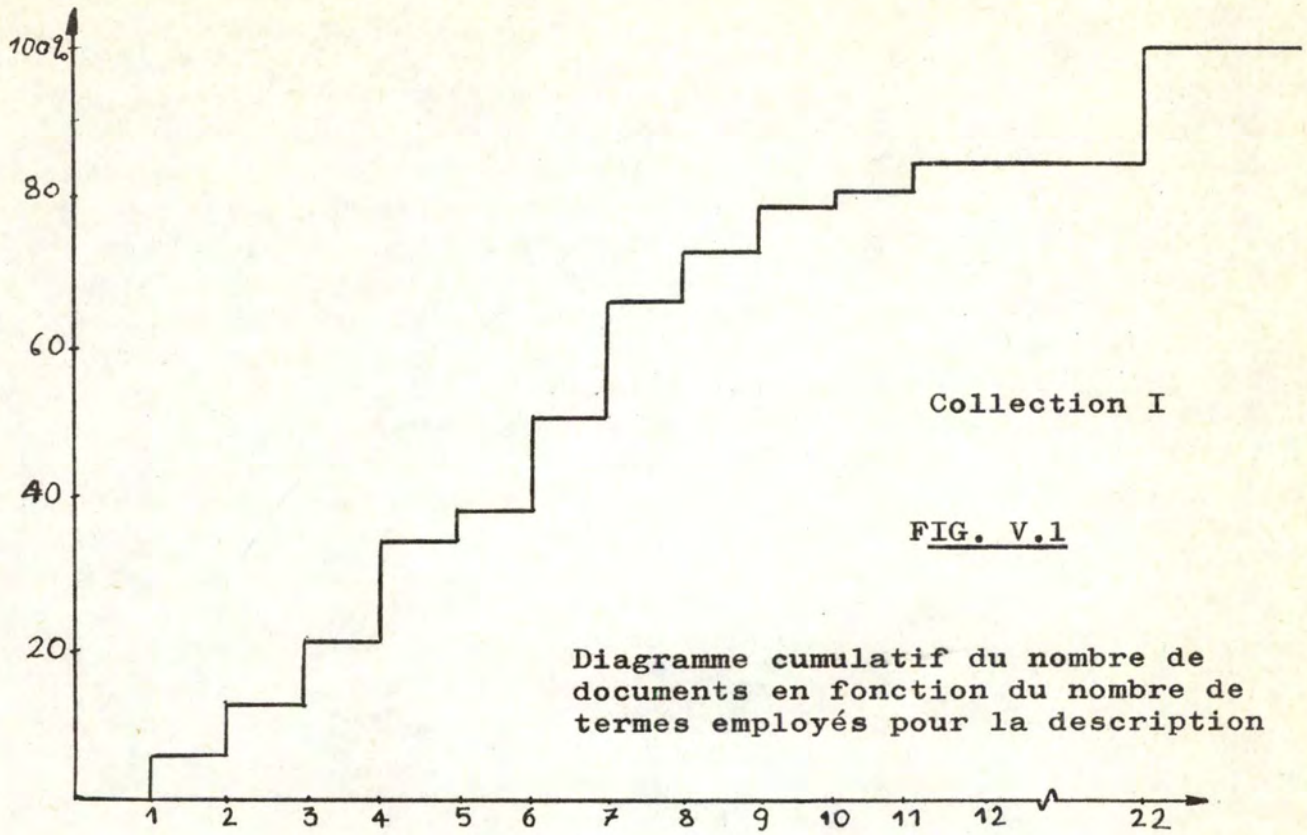
Comme nous serons amenés,par la suite,à faire une étude fréquentielle de la composition des différents thésaurus,nous donnons également deux autres types de renseignements:

1 Diagramme cumulatif du nombre de documents en fonction du nombre de termes qui les dérivent:ce diagramme est donc lié à la complétude de la description des documents

2 Tableau de la répartition des mots clés en fonction de leur fréquence d'emploi:ce tableau met en évidence la spécificité des termes employés.

	1	2	3	4	5	≥6 ≤10	≥11
I	45.5	14.1	11.5	6.8	5.22	10.62	6.26
II	63	19	7.75	4.3	0.0	4.23	1.72
III	57.2	17.9	5.47	5.98	2.98	7.49	2.98
Ib	21.5	23.5	13.1	12.6	9.45	13.05	6.8
IIb	19	37	26.7	7.75	3.45	4.38	1.72

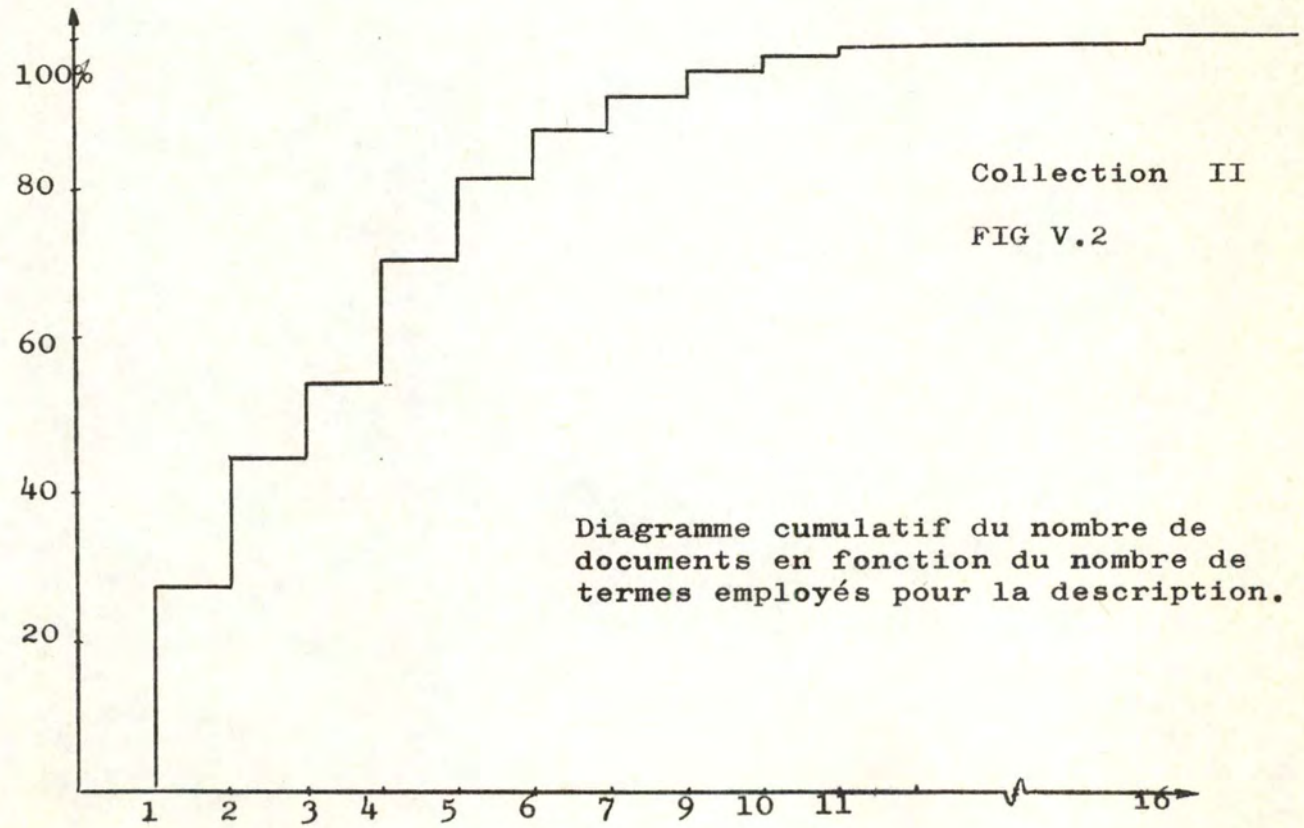
Tableau V.2: répartition(en %) des termes des différentes collections suivant leur fréquence d'emploi.Les deux dernières lignes se rapportent aux collections I et II dont la description est complétée au moyen du thésaurus.



Collection I

FIG. V.1

Diagramme cumulatif du nombre de documents en fonction du nombre de termes employés pour la description



Collection II

FIG V.2

Diagramme cumulatif du nombre de documents en fonction du nombre de termes employés pour la description.

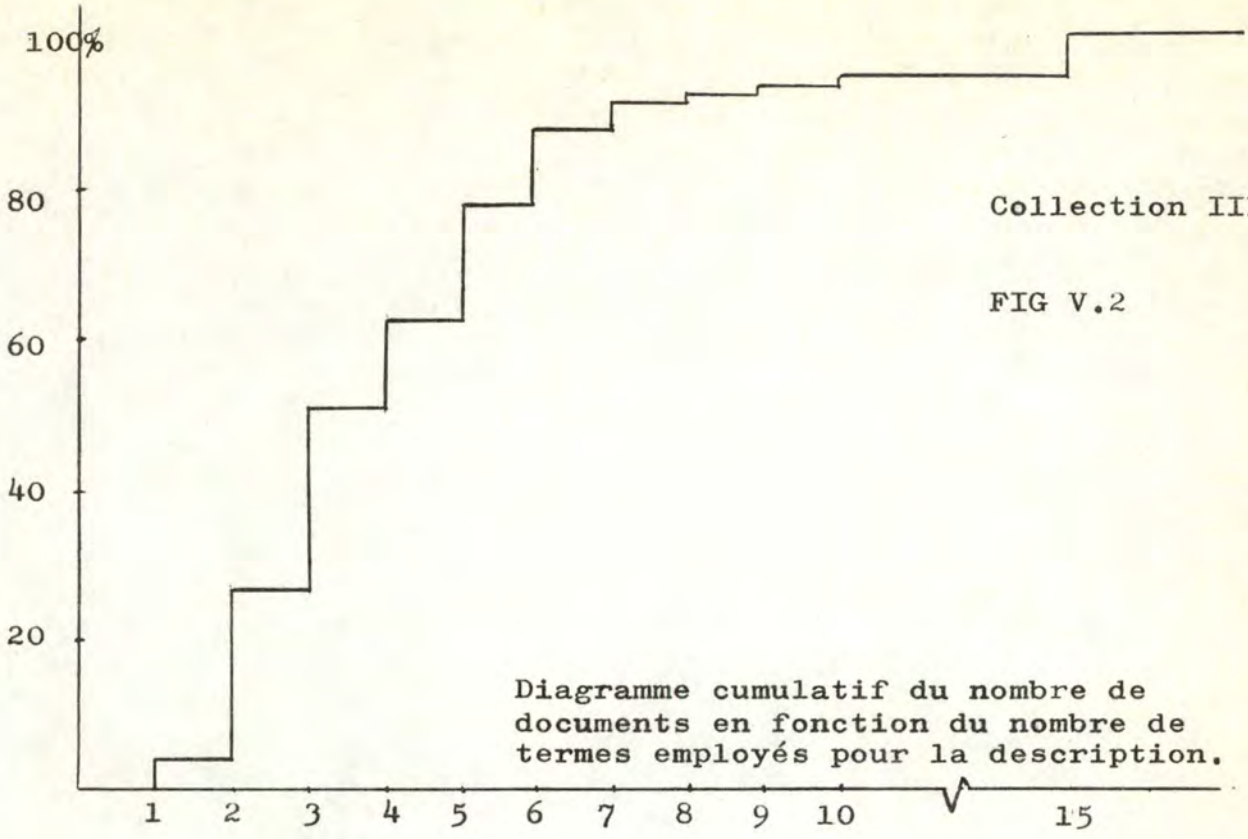


FIG V.2

Diagramme cumulatif du nombre de documents en fonction du nombre de termes employés pour la description.

Conclusion

Les tableaux V.1 et V.2 ainsi que les figures V.1 etV.3 permettent de dégager les prinvipales caractéristiques des trois collections

a La collection I présente la description la plus complète de documents.En effet 12% des documents de cette collection sont décrits par moins de 3 termes et il reste 20% de documents décrits par plus de 10 termes.Pour la collection II, dont la description est la plus élémentaire,ces chiffres sont respectivement 44,5% et 3,27%.

b La collection II est décrite avec les termes les plus spécifiques:82% des termes employés pour l'indexation des documents n'interviennent qu'une ou deux fois contre 59% pour la collection I .

c La collection III se situe entre I et II.

On peut donc s'attendre à ce que la collection I donne lieu à de nombreuses liaisons parasites entre termes lorsqu'on travaille à des seuils de signification très bas,mais que ces liaisons disparaîtront très vite pour des valeurs croissantes du seuil de signification δ.

V.2 Evolution des cliques maximales et composantes connexes en fonction du seuil de signification:étude des paramètres techniques.

Après avoir donné la répartition des cliques maximales et composantes connexes(voir tableaux V.4-V.8),nous examinons pour chacune des trois collections l'évolution des principaux paramètres techniques relatifs aux essaims,en fonction du seuil de signification

- Taille moyenne des cliques maximales et des composantes connexes suivant que les termes isolés(essaims réduits à un seul terme)sont inclus ou non dans le thésaurus figures V.4-V.6 .
- Evolution du nombre d'essaims de taille supérieure à 1: tableau V.9 .
- Diminution du nombre de cliques maximales de taille 1: figure V.7 .
- Nombre d'essaims communs aux deux méthodes;figure V.8 .

Interprétation des résultats

Les tableaux et figures présentés ci-dessous,mettent très nettement en évidence 4 zones du seuil de signification,dans lesquelles les essaims obtenus évoluent d'une manière bien précise,quelle que soit la collection employée.

$$\delta < 0.6$$
$$0.6 \leq \delta \leq 0.7$$
$$0.7 < \delta < 0.75$$
$$\delta \geq 0.75$$

a $\delta < 0.6$:la plupart des courbes évoluent rapidement avec δ.

On distingue un très grand nombre de cliques maximales de taille >1 et au contraire un nombre très petit de composantes connexes et d'essaims réduits à un seul élément.
Pour les composantes connexes,il existe un essaim qui reprend à lui seul environ 75% des termes du vocabulaire employé pour l'indexation.La taille moyenne des cliques maximales(que l'on exclut ou non les essaims réduits à un seul élément)croît modérément.
D'autre part,si l'on examine les courbes obtenues pour les collections I et II,on constate que:

- -La diminution du nombre de cliques maximales(de taille >1) est la plus forte pour la collection I et la plus faible pour la collection II(voir figure V.7)
- -La taille moyenne des cliques maximales(graphique 1 dans les figures V.4-V.6)n'évolue pratiquement pas pour la collection I.

Ces observations s'expliquent comme suit:
Pour de petites valeurs de δ,il existe de nombreuses liaisons peu significatives entre termes.Les liaisons influencent la composition du thésaurus de deux manières.

- -Les essaims comprenant des liaisons fortes entre termes, s'accroissent de termes à liaison faible.
- -Il y a génération de nombreux essaims de taille 2 et 3.

Cette interprétation est la plus évidente pour la collection I qui possède le vocabulaire le moins spécifique:dans ce cas,il existe des liaisons entre la majorité des termes,mais à un niveau faible.

δ \ Taille	1	2	3	4	5	6	7	8	11	14	15	16	17	24	25	119
.5	22	5	3	1			2					1				1
.6	60	9	6	5		1	1	1			1		1		1	
.65	68	9	6	5			1	1		1			1	1		
.7	76	9	5	5			1	1	1			1				
.75	99	11	5	5	5	1	1									
.8	103	9	5	5	5	1	1									
.85	111	11	6	4	4	4	1									

Tableau V.3 :Répartition des composantes connexes
(Collection I)

Tableau V.4:Répartition des cliques maximales
(Collection I)

δ \ Taille	1	2	3	4	5	6	7	8	9	10	11
.5	22	37	31	9	3	9	5	5		1	1
.6	60	26	15	8	5	4	1	2			
.65	68	25	11	7	5	4	1	2			
.7	76	21	10	9	5	3	1	2			
.75	99	16	6	7	4		1				
.8	103	14	6	7	4		1				
.85	111	11	6	4	4	4	1				

δ \ Taille	1	2	3	4	5	6	7	8	9	16	26	28	87
.5	18				1			1					1
.6	28	5	2	2	1	2	1			1		1	
.65	30	4	2	2	1	2	1			1		1	
.7	32	4	2	2	1	2	1			1	1		
.75	57	18	2	3					1				
.8	57	18	2	3					1				
.85	63	15	2	3					1				

Tableau V.5: Répartition des composantes connexes.
(Collection II)

Tableau V.6: Répartition des cliques maximales.
(Collection II)

δ \ Taille	1	2	3	4	5	6	7	8	9	10	11	12
.5	18	12	22	15	2	1	1				2	
.6	28	23	13	5	4					2		
.65	30	22	13	5	4					2		
.7	32	20	13	5	4					2		
.75	57	18	2	3					1			
.8	57	18	2	3					1			
.85	57	18	2	3					1			

δ \ Taille	1	2	3	4	5	6	8	12	26	36	126
.5	33	2	4	3	2	1					1
.6	66	11	10	3	3	2	1	1		1	
.65	71	11	9	3	3	2		1	1		
.7	77	8	9	3	3	2	1	1			
.75	126	15	14	1	1	1					
.8	126	15	14	1	1	1					
.85	129	15	13	1	1	1					
.9	129	15	13	1	1	1					

Tableau V.7: Répartition des composantes connexes.
(Collection III)

Tableau V.8: Répartition des cliques maximales.
(Collection III)

δ \ Taille	1	2	3	4	5	6	7	8
.5	33	53	23	20	8		1	2
.6	66	34	22	14	2		2	
.65	71	28	22	14	2		2	
.7	77	25	29	14	2		2	
.75	126	15	14	1	1	1		
.8	126	15	14	1	1	1		
.85	129	15	13	1	1	1		
.9	129	15	13	1	1	1		

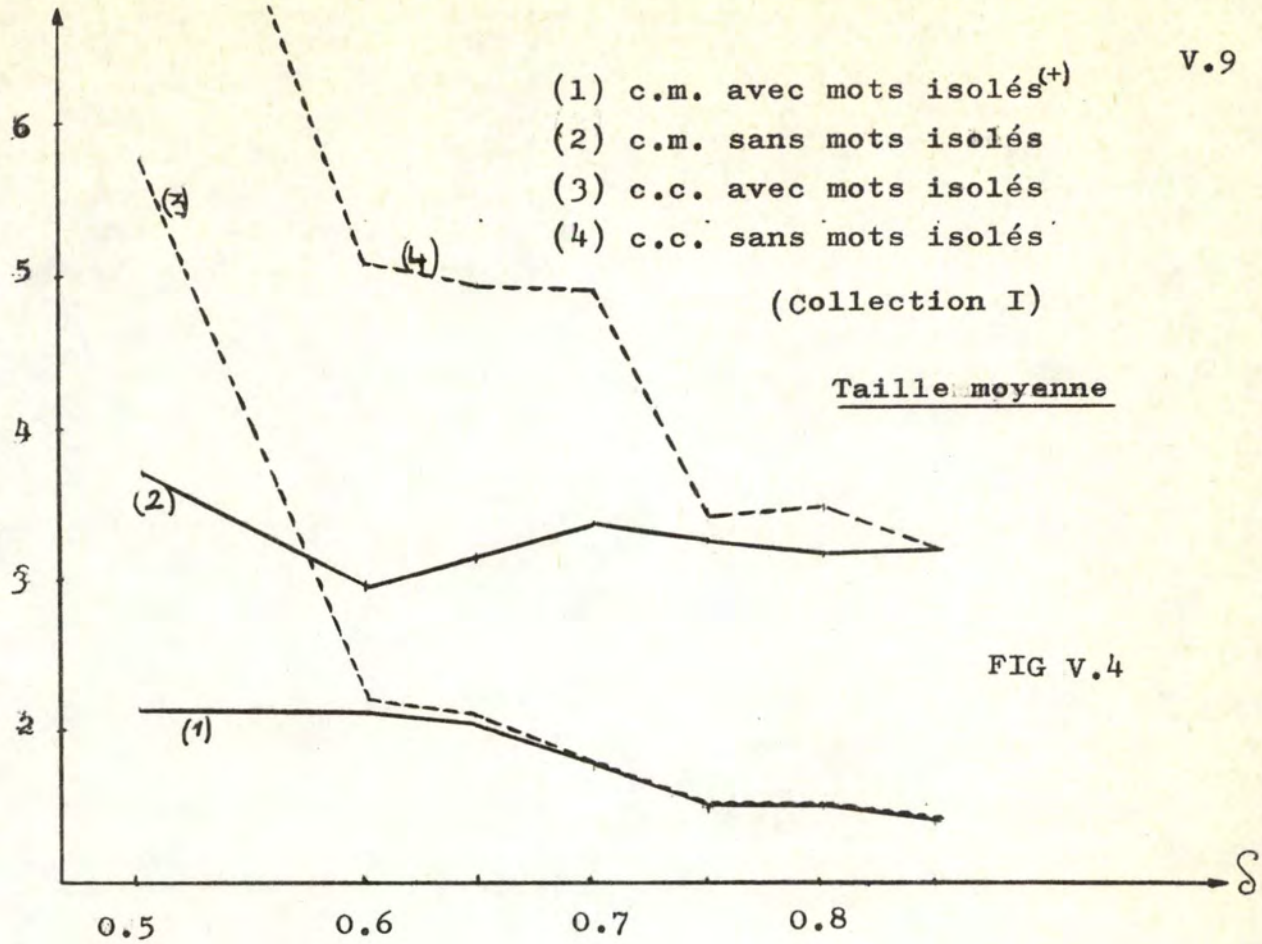


FIG V.4

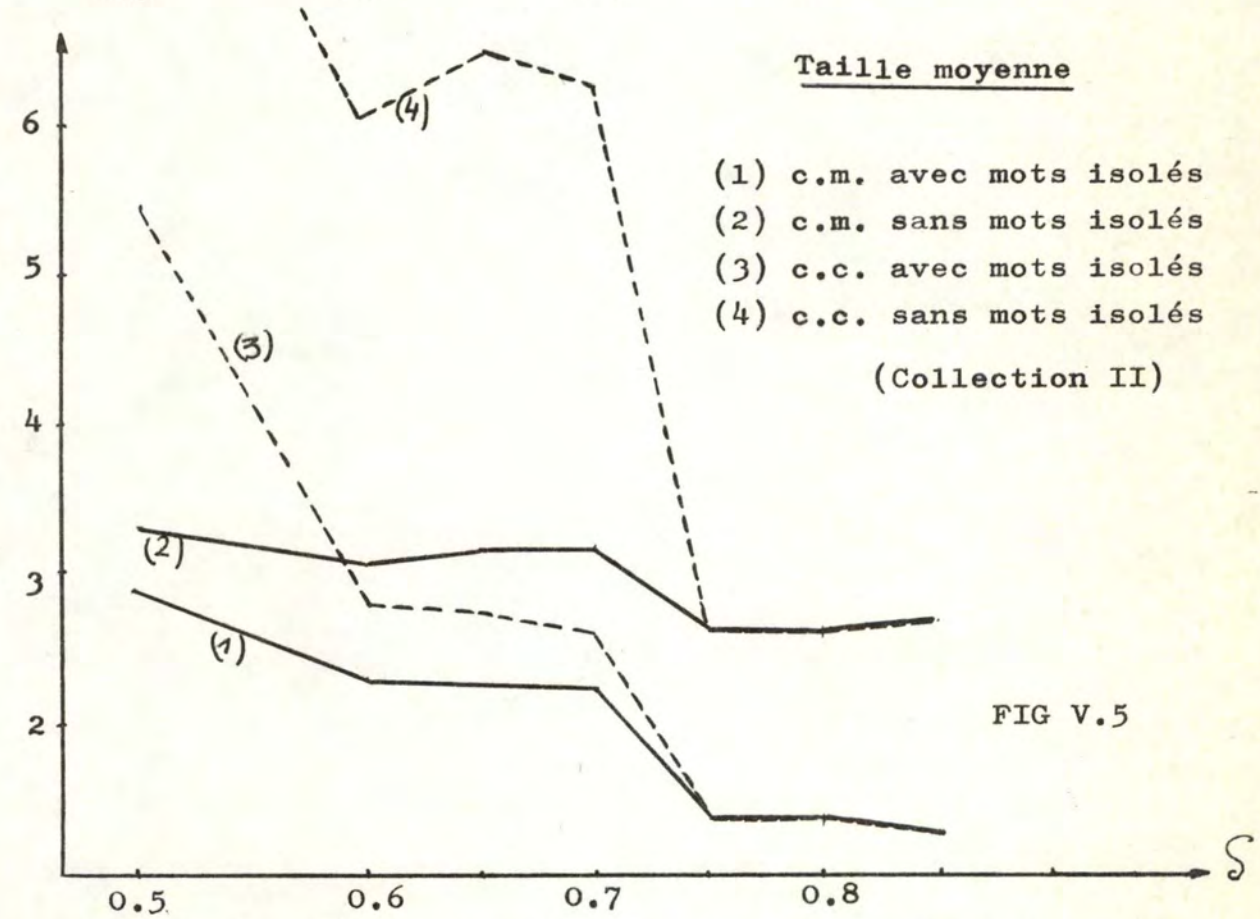


FIG V.5

(+):un mot-clé est dit isolé s'il constitue un essaim à lui seul. (il ne présente donc pas d'intérêt pour la construction de thésaurus)

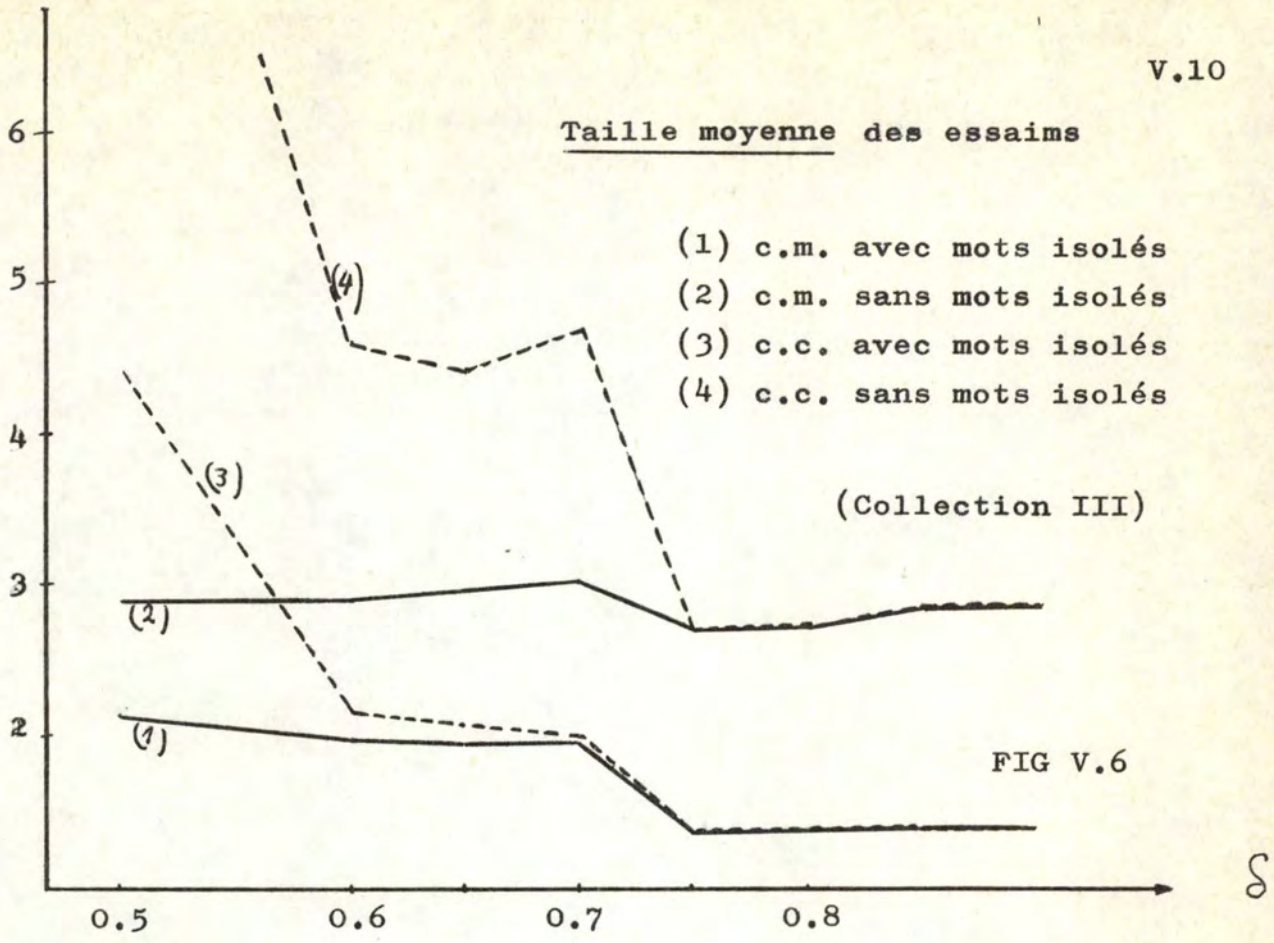
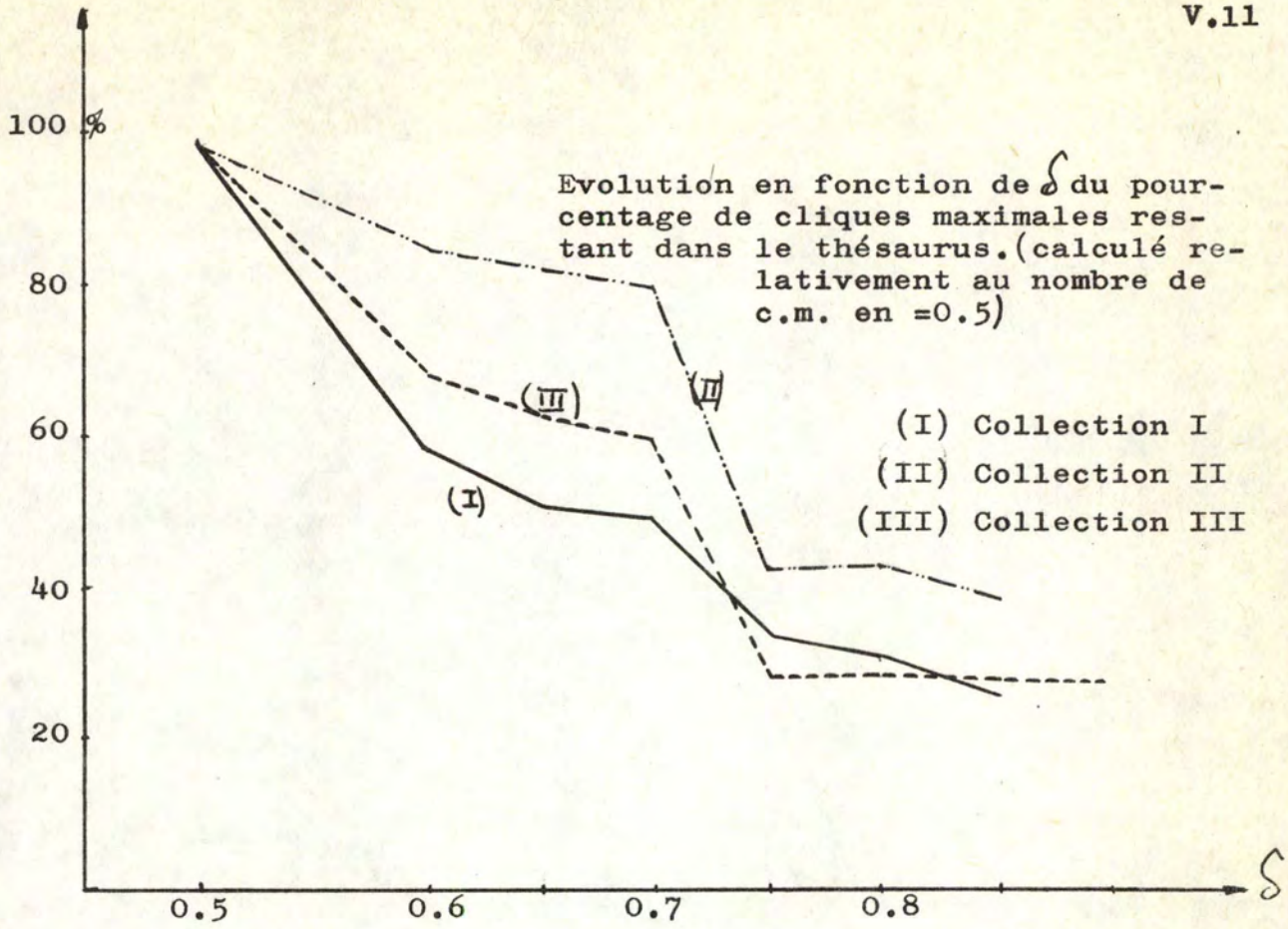


FIG V.6

	I		II		III	
	c.m.	c.c.	c.m.	c.c.	c.m.	c.c.
.5	102	12	57	4	108	13
.6	61	26	47	15	74	32
.65	53	23	46	14	68	32
.7	51	23	44	14	65	29
.75	34	28	24	24	32	32
.8	32	26	24	24	32	32
.85	26	26	21	21	31	31
.9	-	-	-	-	31	31

Tableau V.9: Tableau donnant l'évolution du nombre d'essaims de taille supérieure à 1.
(c.m. =cliques maximales, c.c.=composantes connexes)



Evolution en fonction de δ du pourcentage de cliques maximales restant dans le thésaurus.(calculé relativement au nombre de c.m. en =0.5)

(I) Collection I
(II) Collection II
(III) Collection III

FIG V.7

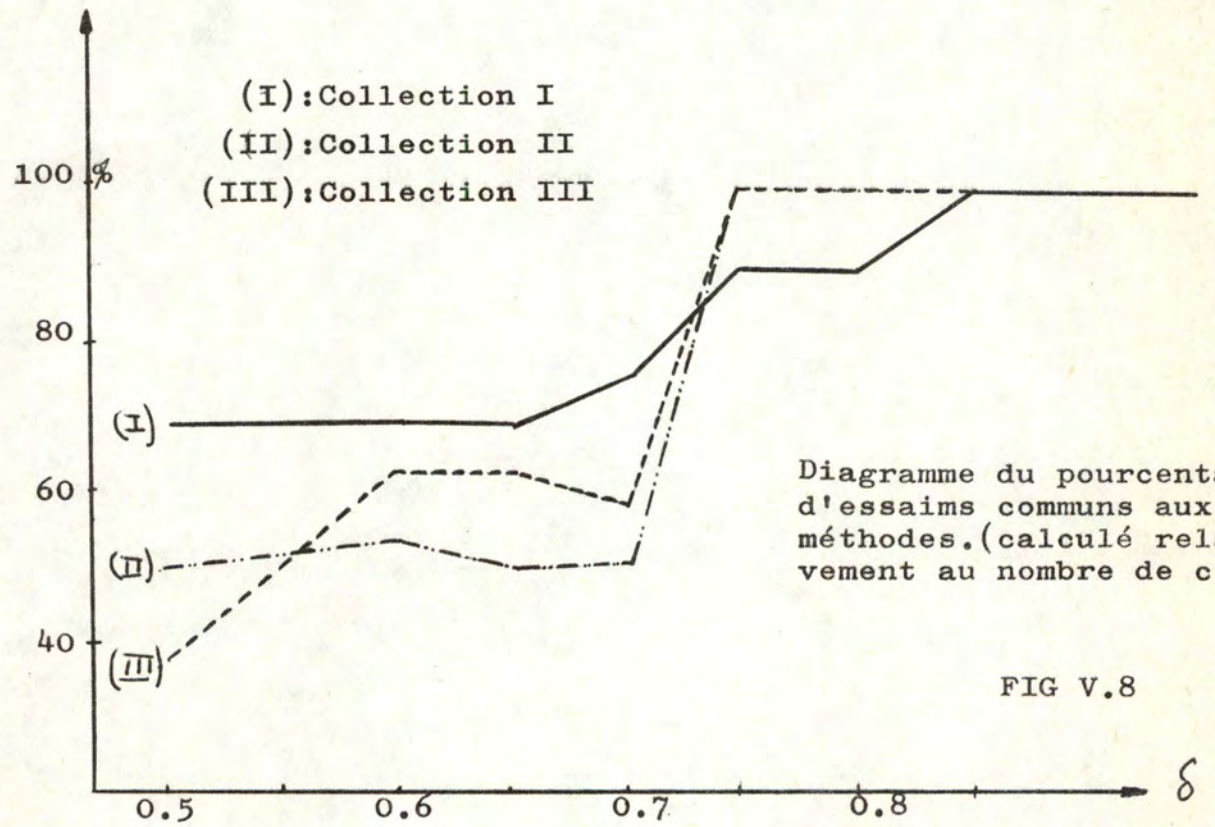


Diagramme du pourcentage d'essaims communs aux 2 méthodes.(calculé relativement au nombre de c.c.

FIG V.8

En conclusion, la zone $\delta < 0.6$ est une zone instable, peu intéressante pour le retrieval (ceci est d'autant plus vrai si l'on considère une évolution dynamique du vocabulaire du thésaurus).

b $0.6 \leq \delta \leq 0.7$: zone de stabilité.

Les tableaux V.3-V.8 indiquent que la répartition des essaims est relativement stable lorsque l'on passe de $\delta = 0.6$ à $\delta = 0.65$ et $\delta = 0.7$. Alors qu'il y avait une diminution (pour la collection I) de 40% du nombre de cliques maximales entre 0.5 et 0.6, elle n'est plus que de 8% entre 0.6 et 0.7; ces chiffres sont respectivement de 15% et 5% pour la collection II et de 30% et 8,5% pour la collection III. De même, il y a une stabilisation pour les composantes connexes et les essaims réduits à un seul élément (voir première colonne, tableaux V.3-V.8). On observe le même phénomène pour les tailles moyennes, quelles que soient les courbes (1), (2), (3), (4), des figures V.4-V.6 .
Cette relative insensibilité des différents paramètres examinés, s'explique par le fait, qu'au delà de 0.6, l'on ne garde plus que des liaisons significatives entre termes. Une fois débarassé des liaisons perturbatrices, le thésaurus devrait évoluer de manière lente et continue pour des valeurs croissantes de δ.
C'est évidemment dans cette zone que nous choisirons le seuil de signification pour la construction automatique d'un thésaurus. D'une part, les thésaurus correspondants respectent bien les principes énoncés au chapitre III. D'autre part, comme on désire promouvoir les possibilités de substitution des termes, on a intérêt à choisir des seuils relativement bas.
Le tableau V.9 montre également qu'entre 0.6 et 0.7, l'effet de chaînage est encore très important; dans cette région, les composantes connexes ne conviennent donc pas pour la construction d'un thésaurus, plusieurs concepts bien précis risquant d'être regroupés dans un même essaim par suite de la signification multiple d'un terme.

c $0.7 < \delta < 0.75$: zone de grande instabilité.

Tous les tableaux et graphiques exposés montrent une cassure très nette des courbes lorsque l'on passe de 0.7 à 0.75.
Nous observons en effet:

1 Une diminution brusque

- de la taille moyenne des essaims (voir courbes des figures V.4-V.6)
- du nombre de cliques (de taille 1): au moins 30%.

2 Une augmentation brusque

- du nombre de termes isolés (1ère colonne des tableaux V.3-V.8)
- du nombre d'essaims identiques, obtenus par les 2 méthodes étudiées (voir figure V.8)

Nous ne pouvons expliquer cette profonde modification du thésaurus à partir des considérations développées jusqu'à présent. Nous donnons au paragraphe suivant une interprétation basée sur

une étude fréquentielle des termes composant le thésaurus.

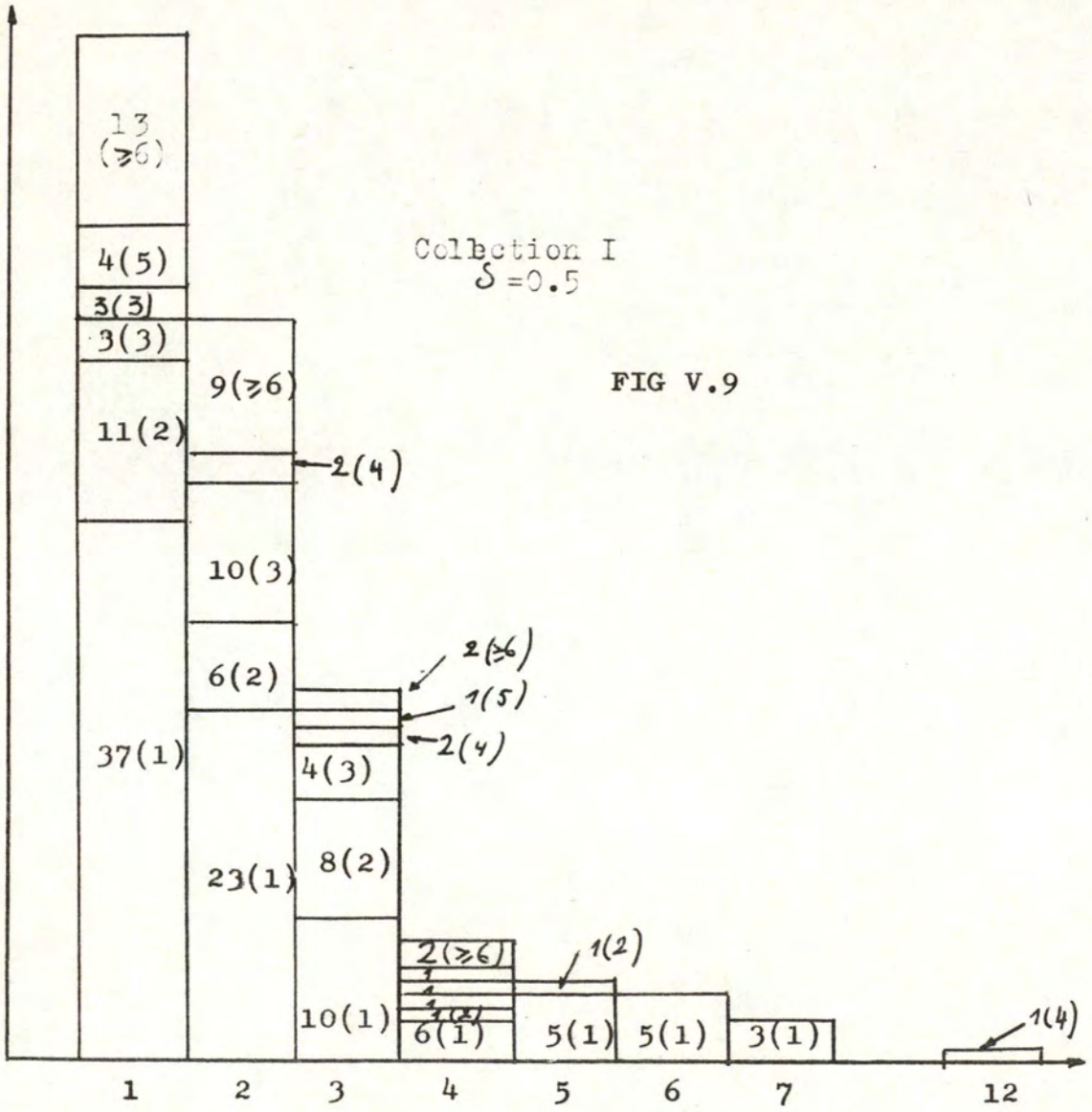
d $\delta \geq 0.75$ zone de grande stabilité.
Tous les paramètres étudiés montrent une très grande stabilité vis àvis du seuil de signification.De plus,pour les collections (II) et (III),les essaims obtenus par les 2 méthodes sont identiques pour $\delta > 0.75$;le même phénomène se produit pour (I) $\delta \geq 0.85$:il n'y a pas d'effet de chaînage,Nous pouvons prévoir que le thésaurus n' évoluera guère entre $\delta = 0.75$ et $\delta = 1.0$.Cette zone n'est toute fois pas utilisée en pratique,car comme nous le verrons au paragraphe suivant,les liaisons exigées entre termes sont tellement fortes qu'il y a peu de possibilités de substitution.

V.3 Evolution des cliques maximales et composantes connexes en fonction du seuil de signification:étude fréquentielle.

L'étude d'un thésaurus,au moyen de paramètres habituellement utilisés tels que taille moyenne et nombre d'essaims,permet de dégager certains principes relatifs à la construction et à l' emploi de thésaurus.Toutefois,elle ne nous apporte pas de renseignements quant à la manière d'indexer les documents d'une collection.C'est pourquoi nous avons entrepris une étude fréquentielle des termes composant le thésaurus.

Les tableaux V.9 à V.17 donnent, pour différentes valeurs du seuil de signification,la composition du thésaurus en fonction de la fréquence d'occurrence des mots clés dans le thésaurus et de la fréquence d'occurrence des mots clés dans le vocabulaire d'indexation. La notation 23(1) dans la 2ème colonne du tableau V.9 signifie que 23 termes de fréquence 1 interviennent 2 fois dans le thésaurus. Nous donnons également le degré de recouvrement des essaims sous forme du pourcentage de termes du thésaurus figurant 1,2,3...fois dans le thésaurus.

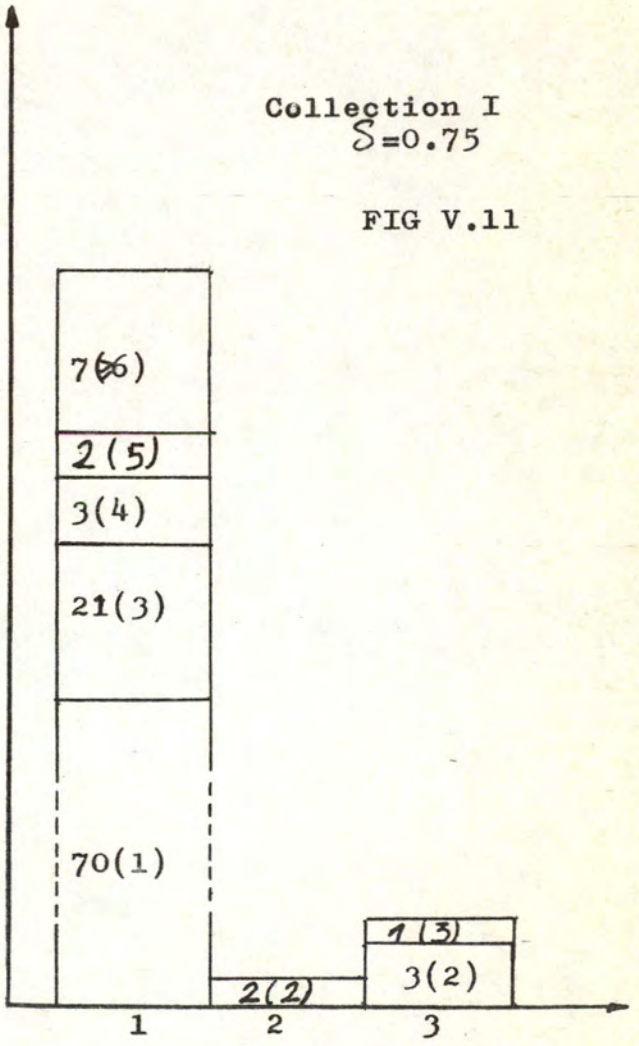
Les figures V.18 à V.20 présentent l'évolution ,en fonction du seuil de signification du pourcentage des termes de fréquence 1,2,3,...restant dans le thésaurus.



Colbction I
δ =0.5

FIG V.9

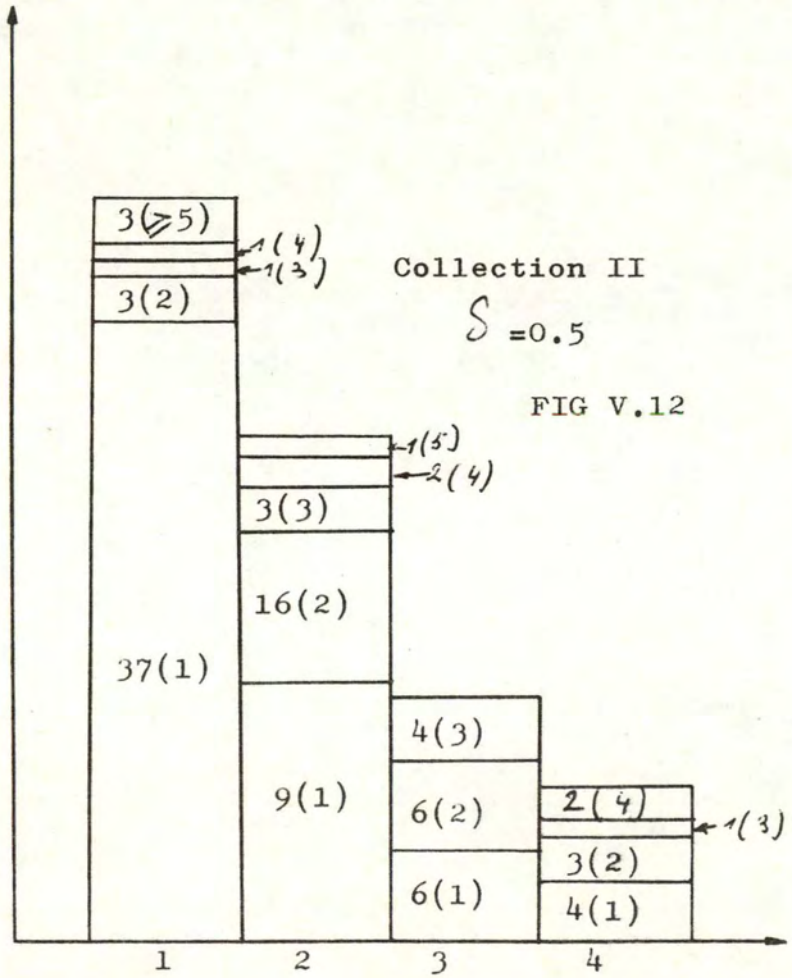
40.5	29.4	15.7	5.23	3.48	2.9	1.73		0.57



Collection I
δ=0.75

FIG V.11

93.7	2.1	4.22

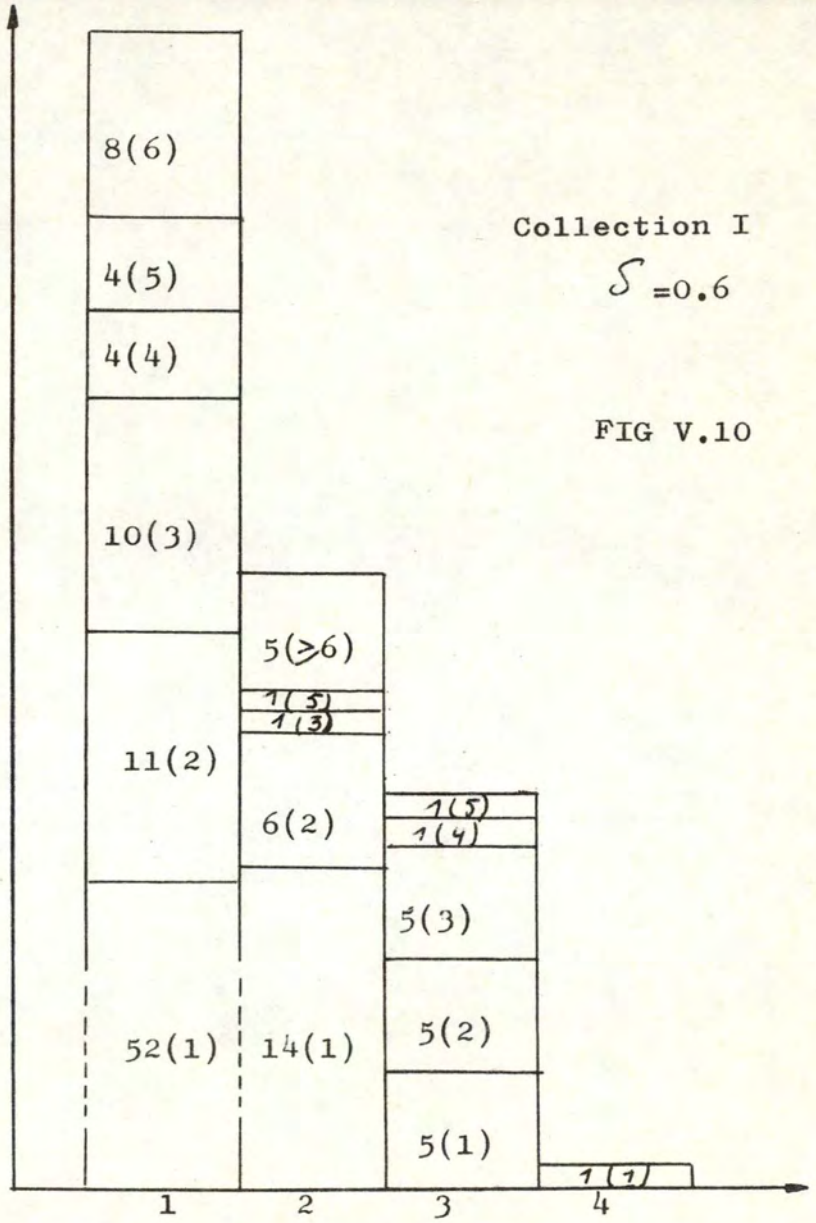


Collection II

δ =0.5

FIG V.12

44.1	30.4	15.7	9;8



Collection I

δ =0.6

FIG V.10

66.4	20.2	12;65	0.75

Collection II

δ=0.65

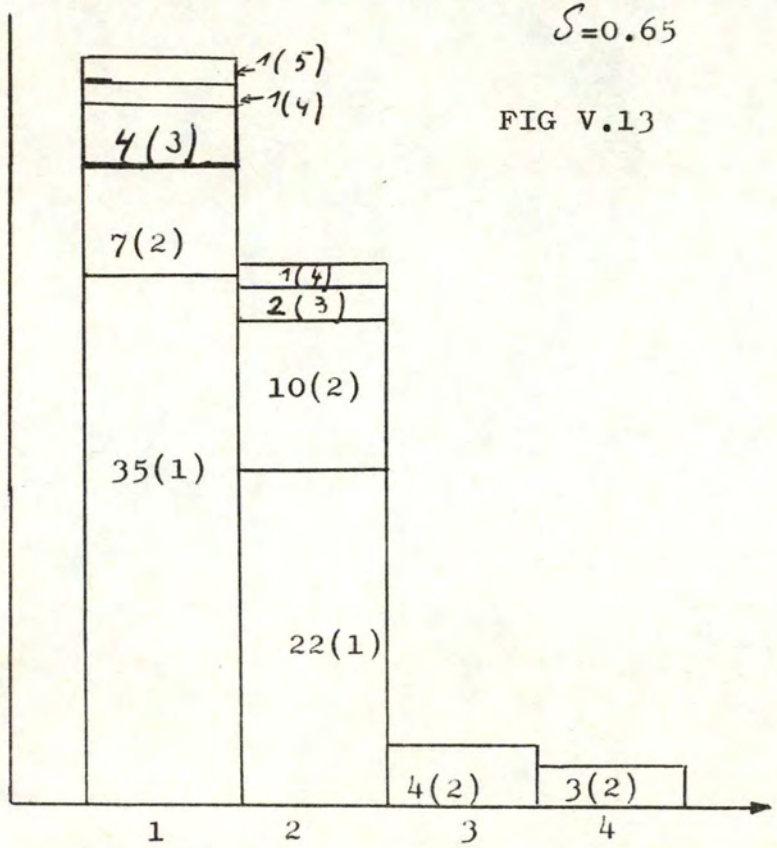


FIG V.13

53.3	38.8	4.5	3.4

Collection III

δ=0.5

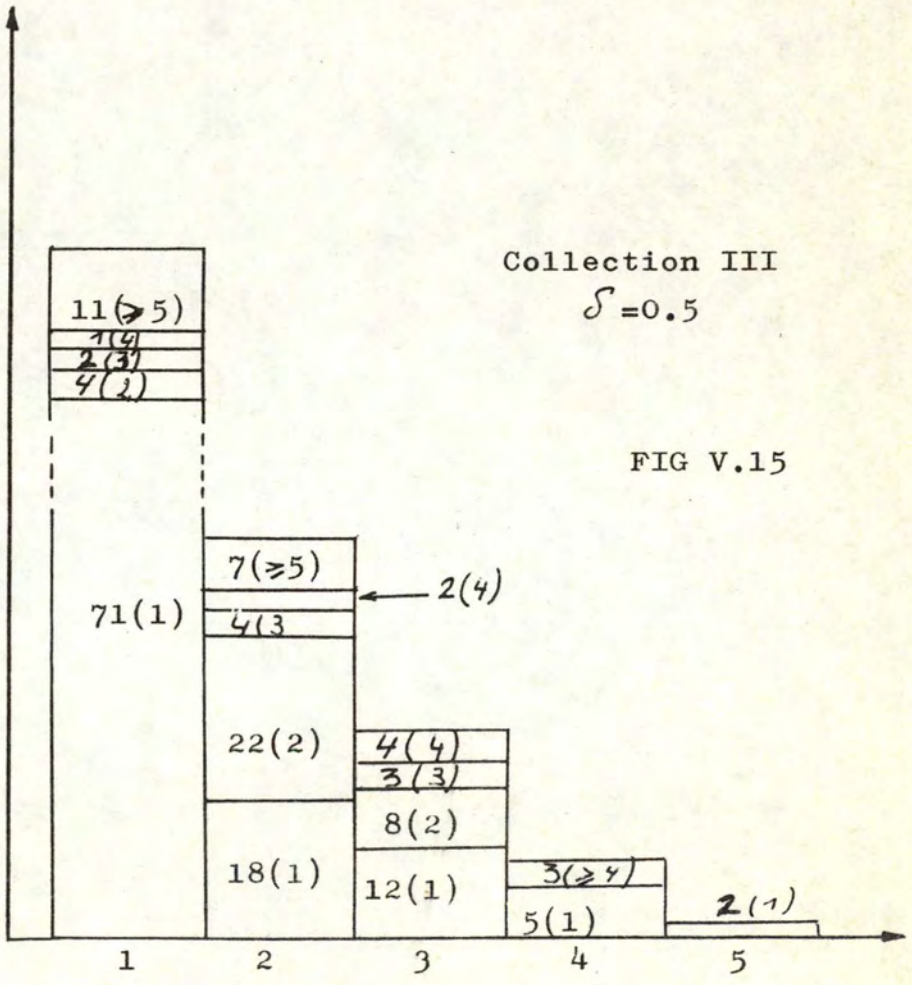
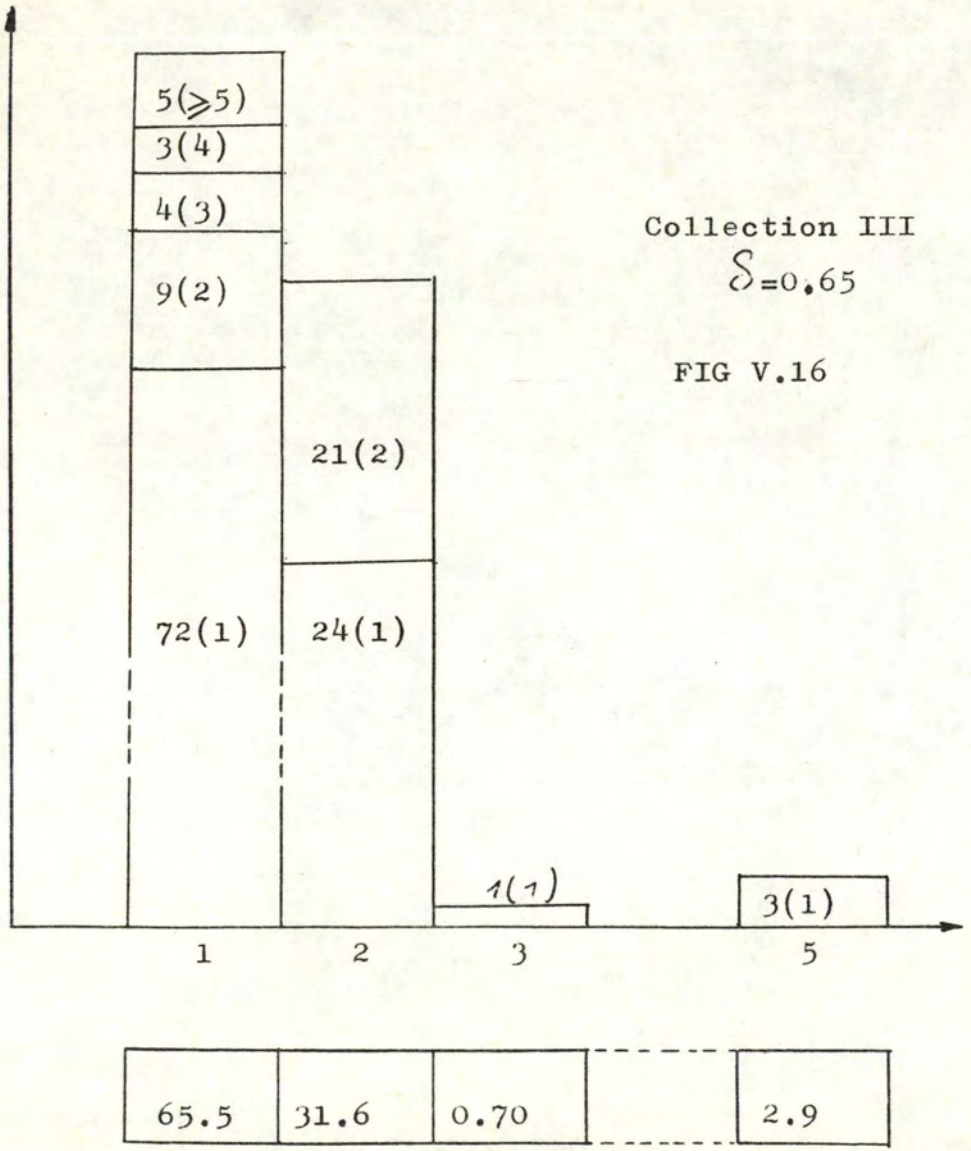


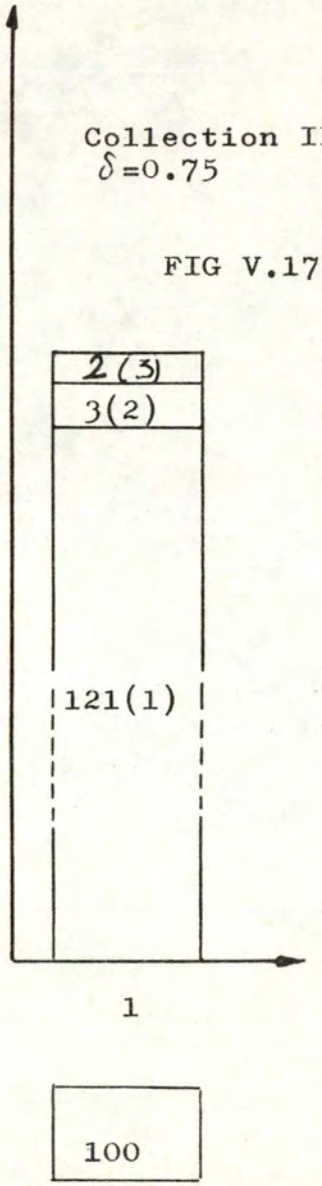
FIG V.15

49.5	29.4	15.0	5.0	1.1



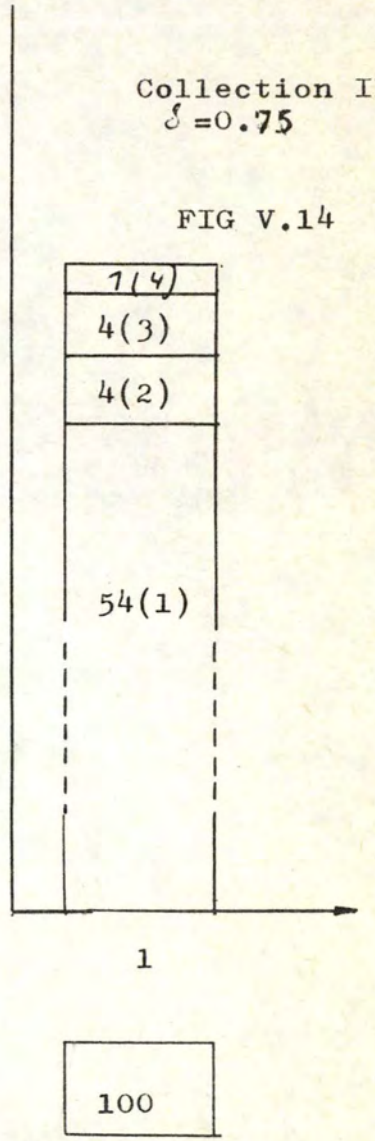
Collection III
δ=0.65

FIG V.16



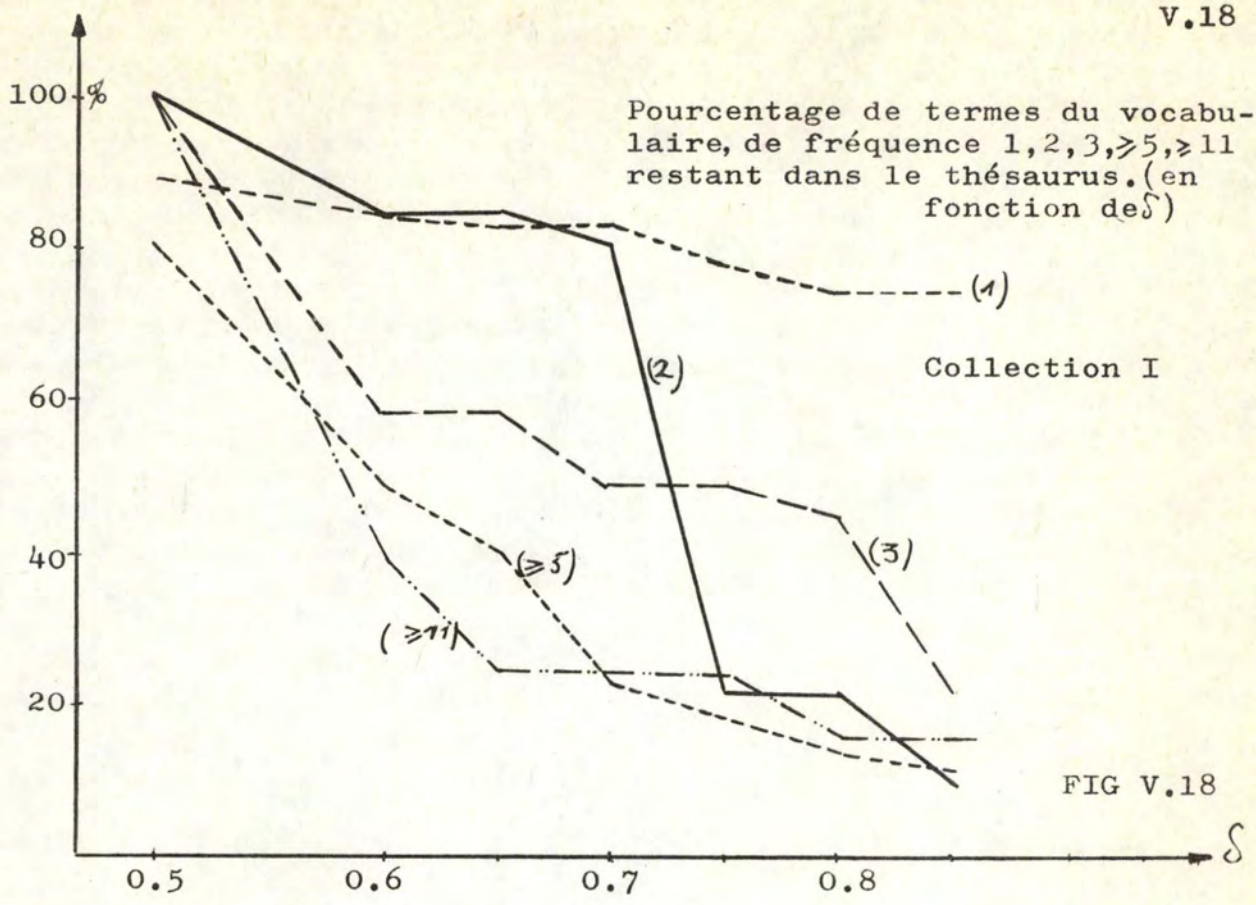
Collection III
δ=0.75

FIG V.17



Collection II
δ=0.75

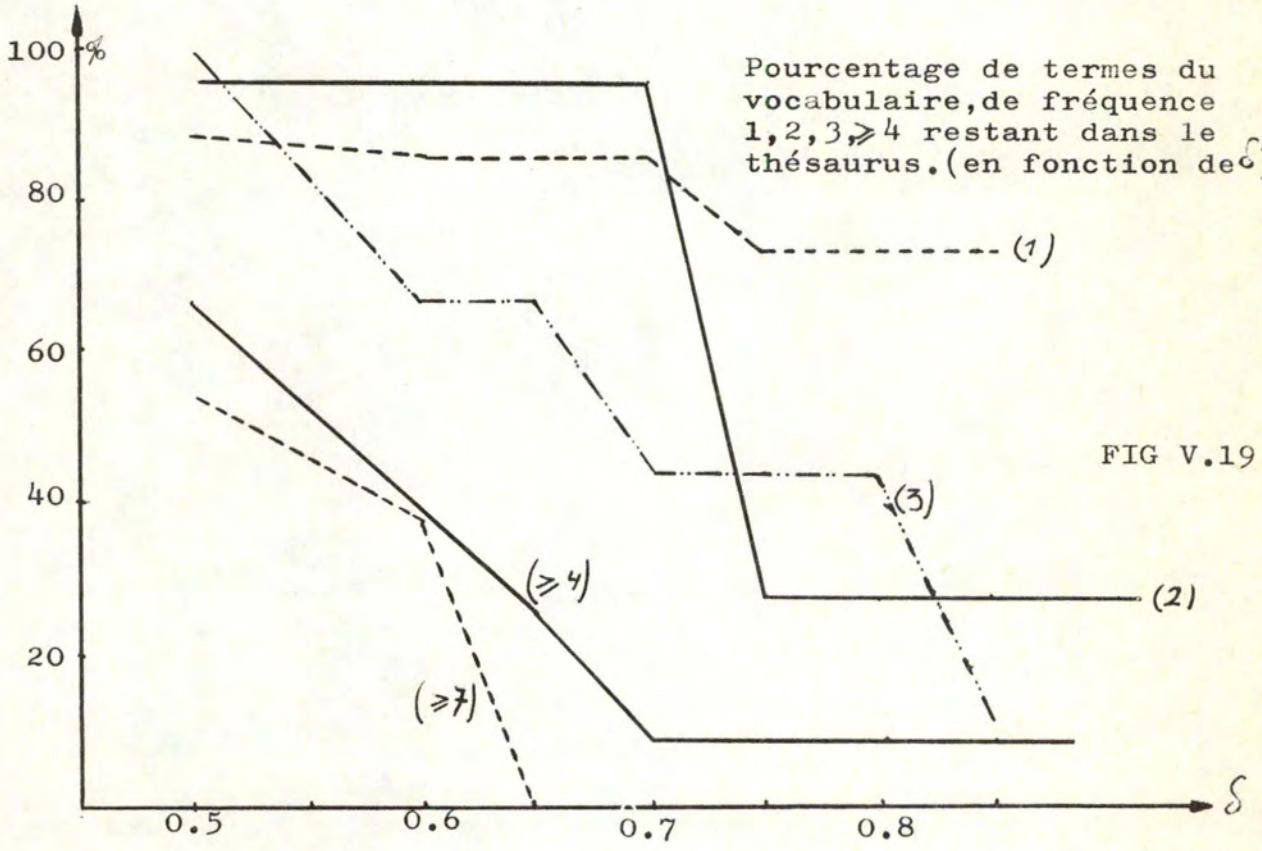
FIG V.14



Pourcentage de termes du vocabulaire, de fréquence 1, 2, 3, ≥5, ≥11 restant dans le thésaurus. (en fonction de δ)

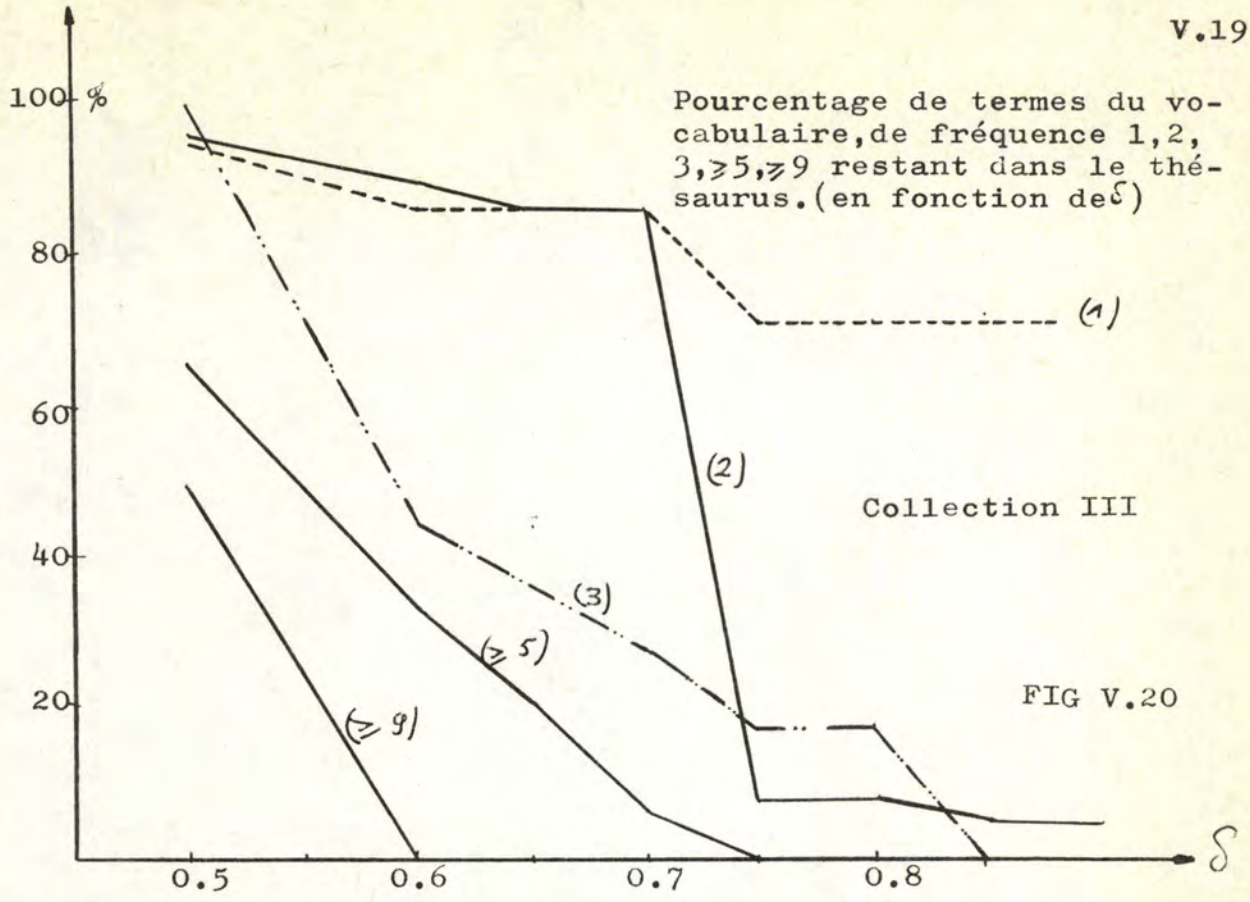
Collection I

FIG V.18



Pourcentage de termes du vocabulaire, de fréquence 1, 2, 3, ≥4 restant dans le thésaurus. (en fonction de δ)

FIG V.19



Pourcentage de termes du vocabulaire, de fréquence 1, 2, 3, ≥5, ≥9 restant dans le thésaurus. (en fonction de δ)

Collection III

FIG V.20

Interprétation

L'examen des histogrammes V.9 à V.17 met en évidence (comme on s'y attendait intuitivement) que la proportion des termes figurant plusieurs fois dans le thésaurus diminue pour des valeurs croissantes de δ. Par exemple, la proportion des termes du thésaurus figurant plus d'une fois dans le thésaurus est de 59.5% au niveau $\delta = 0.5$, 23.6% au niveau 0.6 et 6.3% au niveau 0.75 pour la collection I. Ces chiffres sont respectivement de 55.9, 46.7, 0% pour la collection II et de 50.5, 34.5 et 0% pour la collection III.
Les histogrammes montrent également que parmi les termes du thésaurus, figurent en majorité des termes de fréquence 1 et 2: ce fait est mieux exprimé par les figures V.18 à V.20. De ces graphiques, nous pouvons en effet dégager les constatations suivantes:

a En dehors de l'intervalle $[0.7, 0.75]$, les courbes relatives aux termes de fréquence 1 et 2 sont insensibles aux fluctuations de

b Les courbes de fréquence 1 et 2 présentent une cassure très nette pour 2 lorsqu'on passe de 0.7 à 0.75: il y a en moyenne perte de 10% de termes de fréquence 1 et 60% de termes de fréquence 2. Ces considérations sont surtout valables pour les collections II et III.

c Les termes de fréquence > 2 ne présentent pas de caractéristique particulière pour $\delta \in [0.7, 0.75]$. La proportion de ces termes décroît plus ou moins régulièrement jusqu'à atteindre une valeur limite. Plus les termes ont une fréquence

élevée,plus ils ont tendance à disparaître rapidement du thésaurus.

La constatation b s'explique comme suit:Nous savons que la proportion de termes de fréquence 1 et 2 est très importante(53% pour I et plus de 70% pour II et III).En passant de 0.7 à 0.75,nous détruisons précisément toutes les connexions entre termes de fréquence 1 et termes de fréquence 2.
Supposons qu'on ait la matrice documents-termes suivante:

	t1	t2	t3	t4	t5	t6	t7	t8	t9
D1	1	1		1	1		1	1	1
D2	1		1		1	1	1	1	
D3							1		
D4								1	1

L'application du coefficient de corrélation cosinusoïdale donne par exemple

$S(t1,t2)=0.707$; $S(t2,t3)=0$; $S(t2,t4)=1$; $S(t1,t5)=1$;

$S(t4,t5)=0.707$; $S(t3,t6)=0.707$.

Cette hypothèse se trouve effectivement confirmée lorsqu'on examine les essaims pour $\delta=0.7$ et $\delta=0.75$

La constatation c provient du fait suivant:Au fur et à mesure que δ augmente,ne figurent dans le thésaurus que des termes qui ont tendance à survenir un même nombre de fois dans les mêmes documents: Nous pouvons de nouveau expliquer à partir de la matrice ci-dessus: On a par exemple:

$$S(t1,t2)=0.707;\ S(t1,t7)=\frac{2}{\sqrt{6}}=0.815;\ S(t7,t8)=\frac{2}{3}=0.66$$

Il se fait que,généralement,un terme de fréquence élevée,est employé dans des contextes différents et donc dans des documents différents Nous avons par exemple relevé dans les 3 collections les termes suivants,à usage multiple:fonctions-équations linéaires-programmation-operating systems-processus stochastiques.

Parmi les essaims de termes de fréquence ≥ 3,nous avons relevé

1: Equations différent. elliptiques(5)-Equations différent. paraboliques(5)-Equations différent. hyperboliques(4)

2: Relations(4)-Treillis(3).

3: Ordre complet(3)-Ordre partiel(3).

4: Normes vectorielles(8)-Normes matricielles(9).

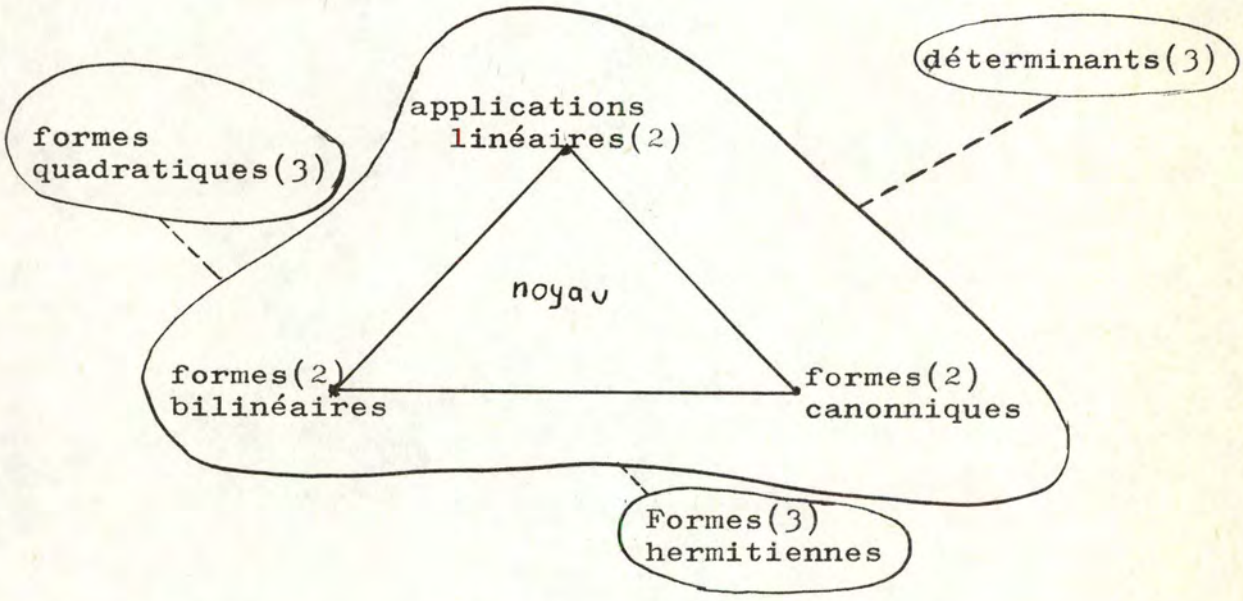
5: Valeurs propres(26)-Vecteurs propres(20).

Les nombres entre parenthèses indiquent la fréquence des termes.

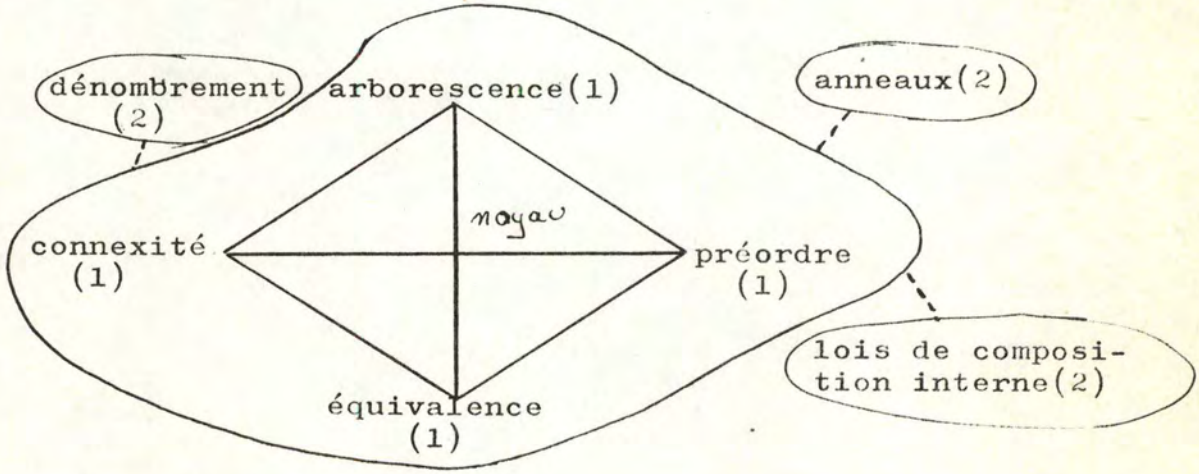
Composition fréquentielle des essaims du thésaurus.

Dans la zone intéressante du seuil de signification($0.6 \leq \delta \leq 0.7$), le thésaurus se compose le plus souvent d'essaims qui se recouvrent partiellement; ce recouvrement est dû au fait que l'on groupe dans un même essaim certains termes dont la fréquence diffère de 1.

Ex. Dans la collection I, au niveau 0.8, nous distinguons 3 essaims à noyau commun.



Au niveau 0.85 ne subsiste plus que le noyau.
Pour la collection I, au niveau 0.7, nous avons encore



La matrice documents-termes relative à ces termes pourrait être

	t1	t2	t3	t4	t5	t6	t7
D1	1	1	1	1	1	1	1
D2					1		
D3						1	
D4							1

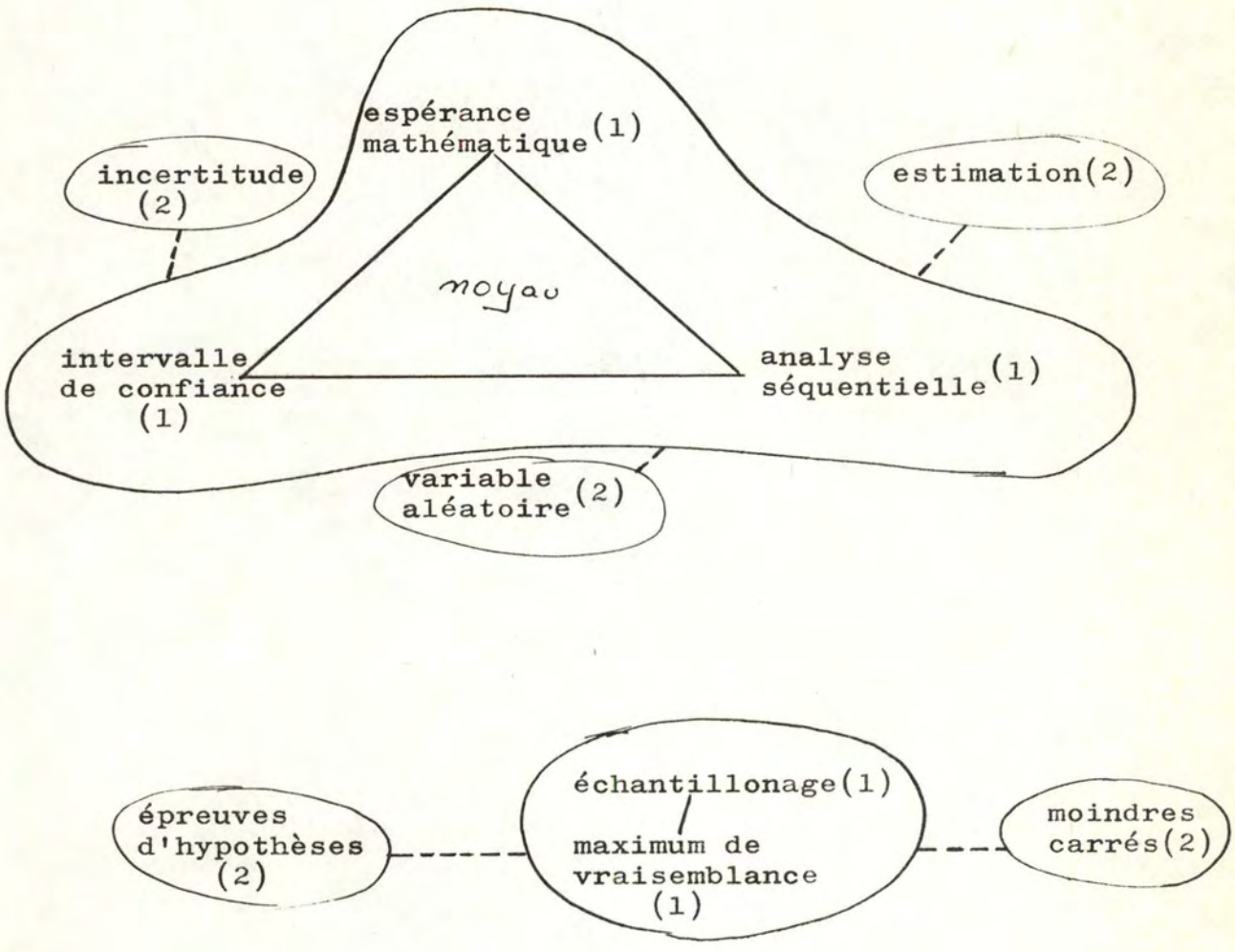
t1,t2,t3,t4=noyau commun

t5=anneaux

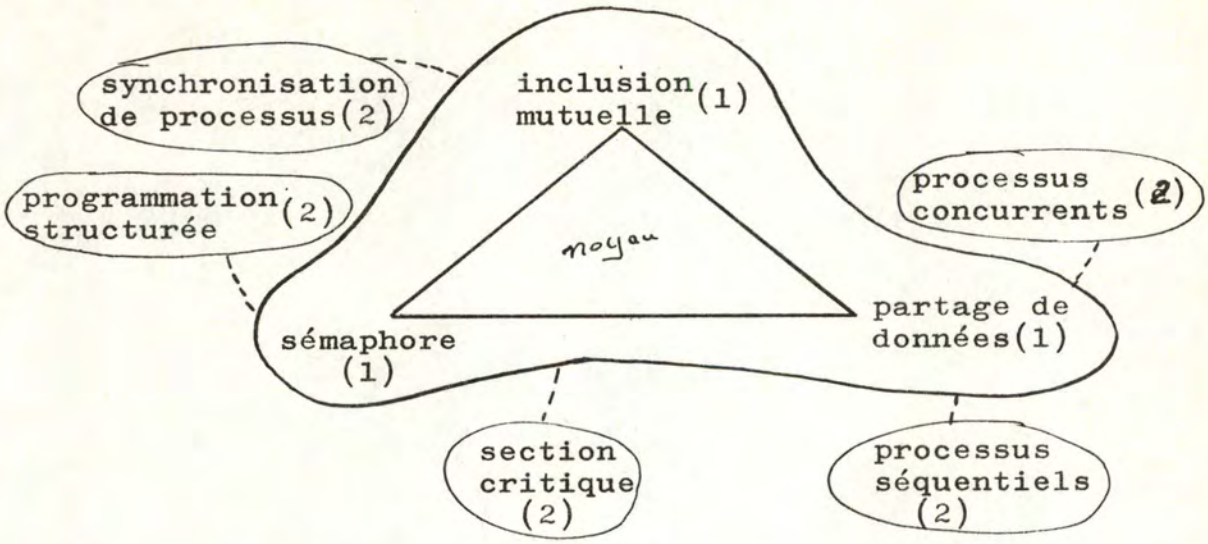
t6=lois de composition interne

t7=dénombrement

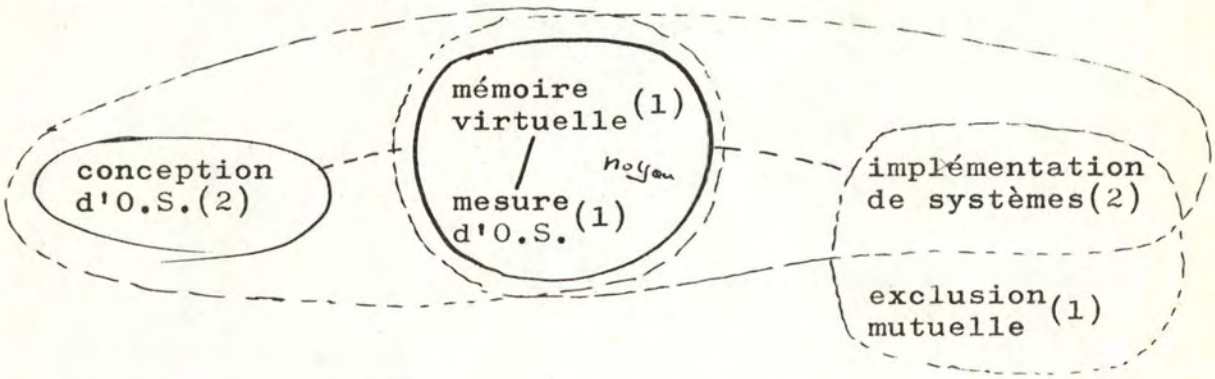
Ceci nous explique pourquoi un terme de fréquence 1 peut survenir plusieurs fois dans le thésaurus.
Pour la collection II,on aurait par exemple:



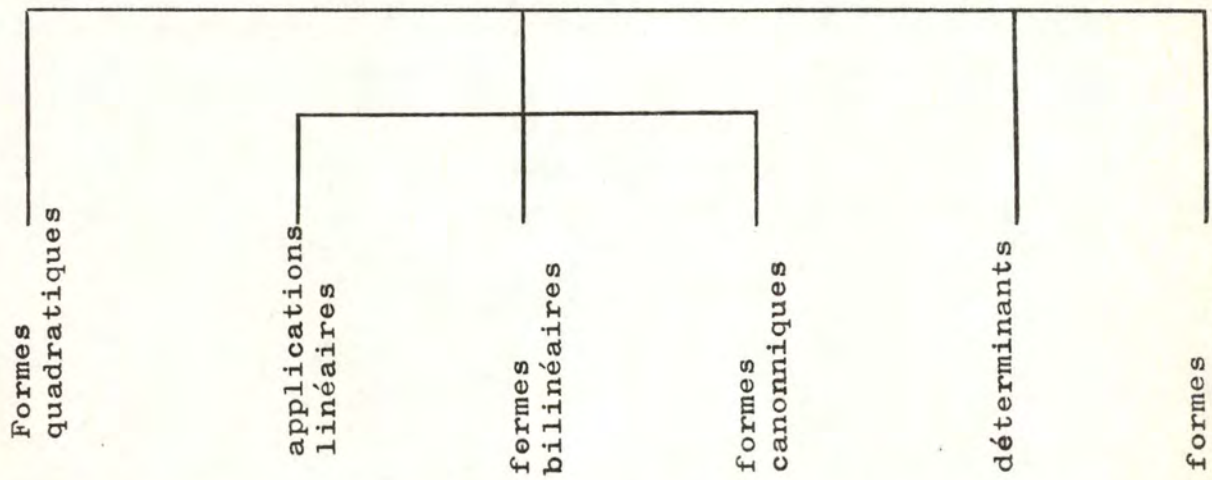
Pour la collection III,on aurait par exemple:



Nous voyons ici l'existence de 5 essaims qui ont en commun 3 termes de fréquence 1.
Tous les essaims ne se recouvrent pas de cette façon:beaucoup ne se recouvrent que par un seul élément : Par exemple,pour la collection III,on aurait



Remarquons enfin que,pour des valeurs de seuil élevées(0.75), nous pouvons entreprendre la même étude à partir du dendrogramme des mots clés.Pour l'essaim d'analyse numérique,on aurait par ex.



V.4 Essaimage des documents

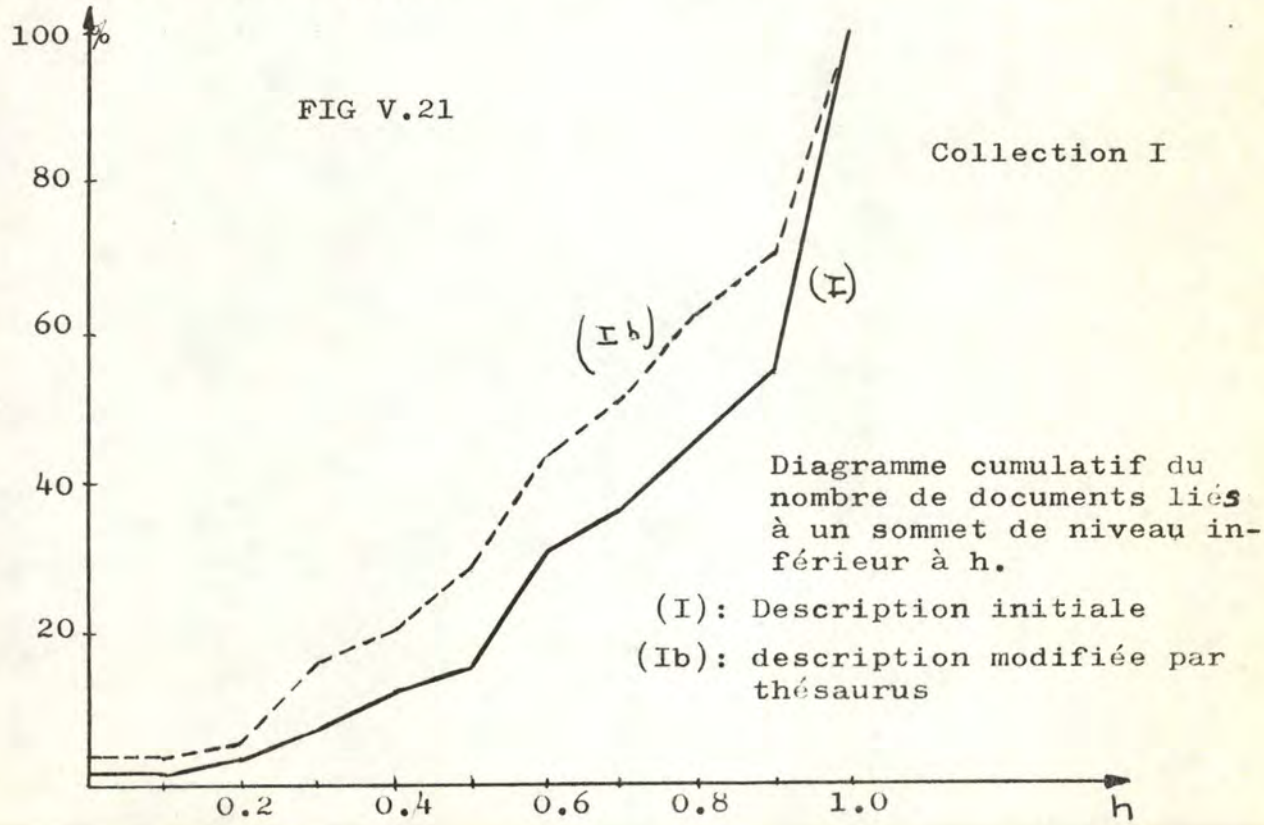
Afin d'éviter, à chaque requête, le balayage séquentiel de l'ensemble des documents, nous avons construit une arborescence des documents déduite du dendrogramme des documents. Nous avons caractérisé la forme du dendrogramme au moyen de 2 diagrammes:

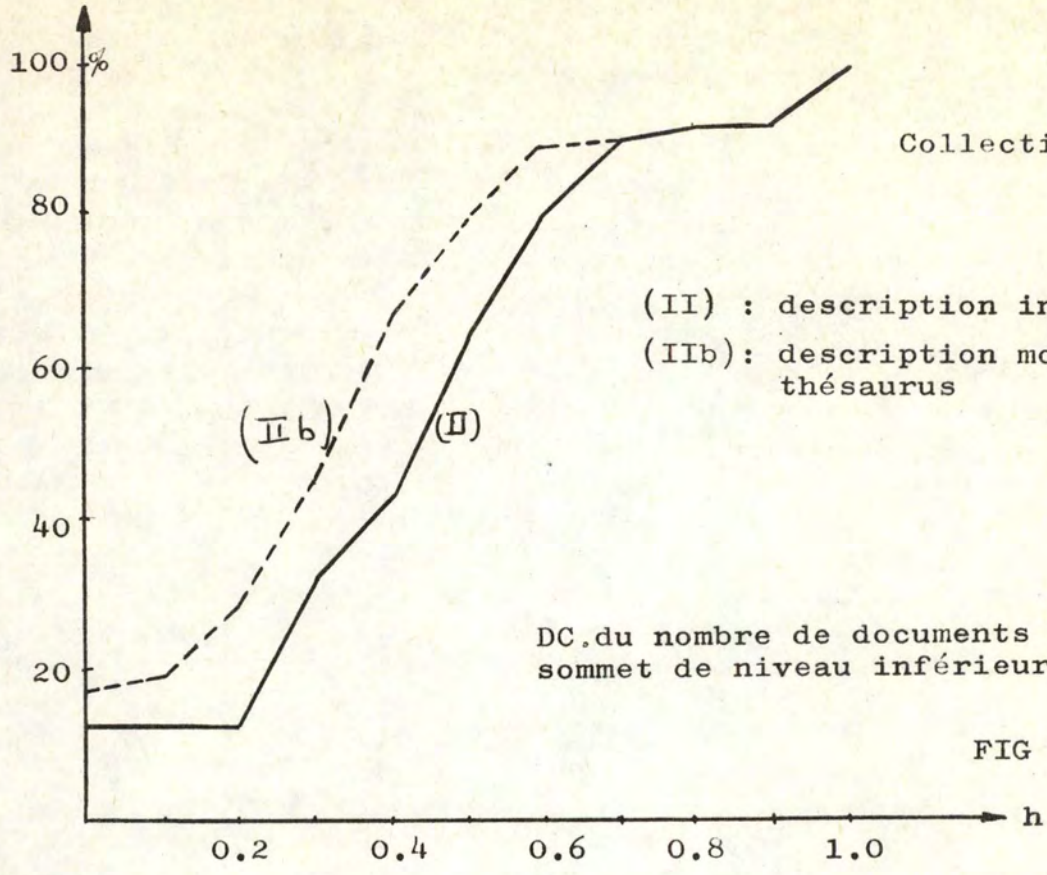
1 Le nombre de documents reliés à un sommet de niveau h où h est une valeur du coefficient de dissimilarité (figures V.21 à V.23)

2 La réduction du nombre de branches pour les valeurs croissantes du coéfficient de dissimilarité (figure V.24 à V.25)

L'examen des figures V.21 à V.25 montre que l'essaimage des documents de la collection II diffère nettement des 2 autres: par exemple, 42% des documents de la collection II sont essaimés au niveau 0.4, 90% au niveau 0.7 et 92% au niveau 0.9.
Ces chiffres sont respectivement de 12,37 et 53% pour la collection I et de 32.6, 45 et 52.4% pour la collection III.
La collection II diffère donc des deux autres par le fait que la description des documents de I et III est beaucoup plus individualisée que pour II: près de la moitié des documents de I et III sont directement reliés au sommet du dendrogramme.
Nous avons relevé le 1er paramètre pour les dendrogrammes des mots-clés des 3 collections, afin de voir s'il existait une relation évidente entre les classifications des documents et des mots-clés d'une collection. La figure V.26 montre que les grandeurs calculées précédemment sont en moyenne 73, 95, 98.5%, les 3 courbes relatives aux 3 vocabulaires d'indexation présentant plus ou moins les mêmes caractéristiques. La collection II possède, toutefois, à nouveau l'essaimage le plus fin.

FIG V.21





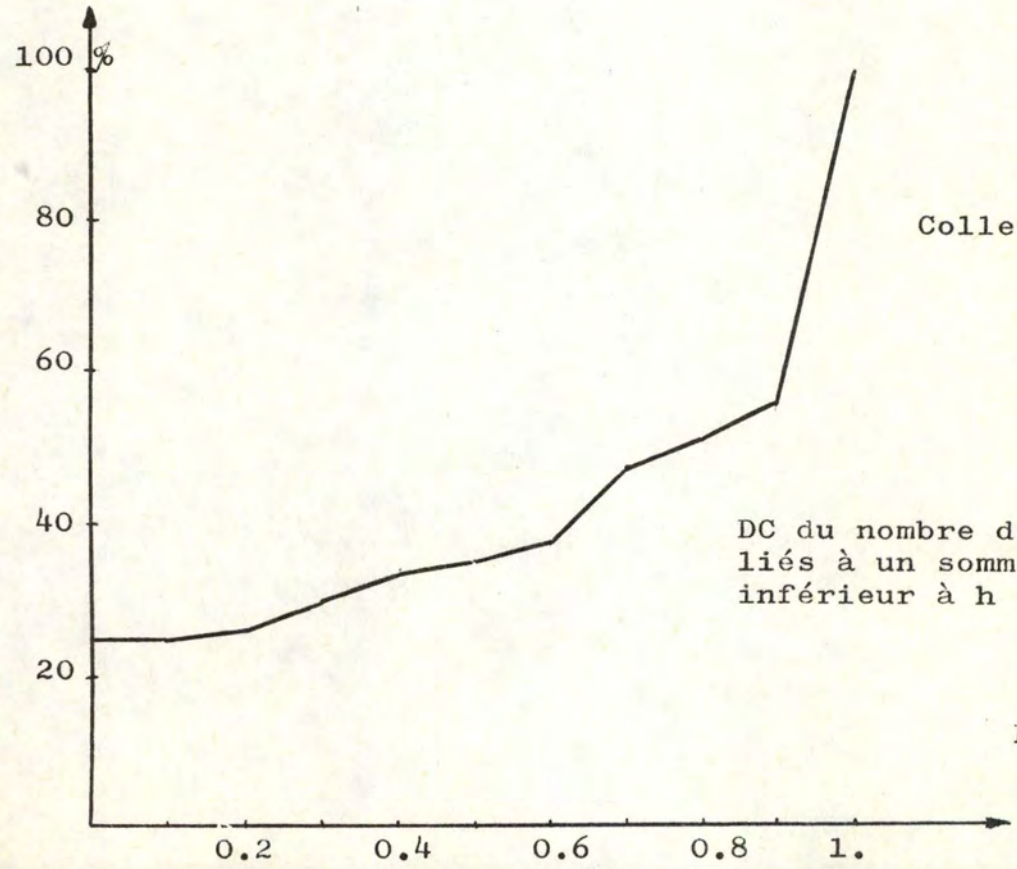
Collection II

(II) : description initiale

(IIb): description modifiée par thésaurus

DC.du nombre de documents liés à un sommet de niveau inférieur à h

FIG V.22



Collection III

DC du nombre de documents liés à un sommet de niveau inférieur à h

FIG V.23

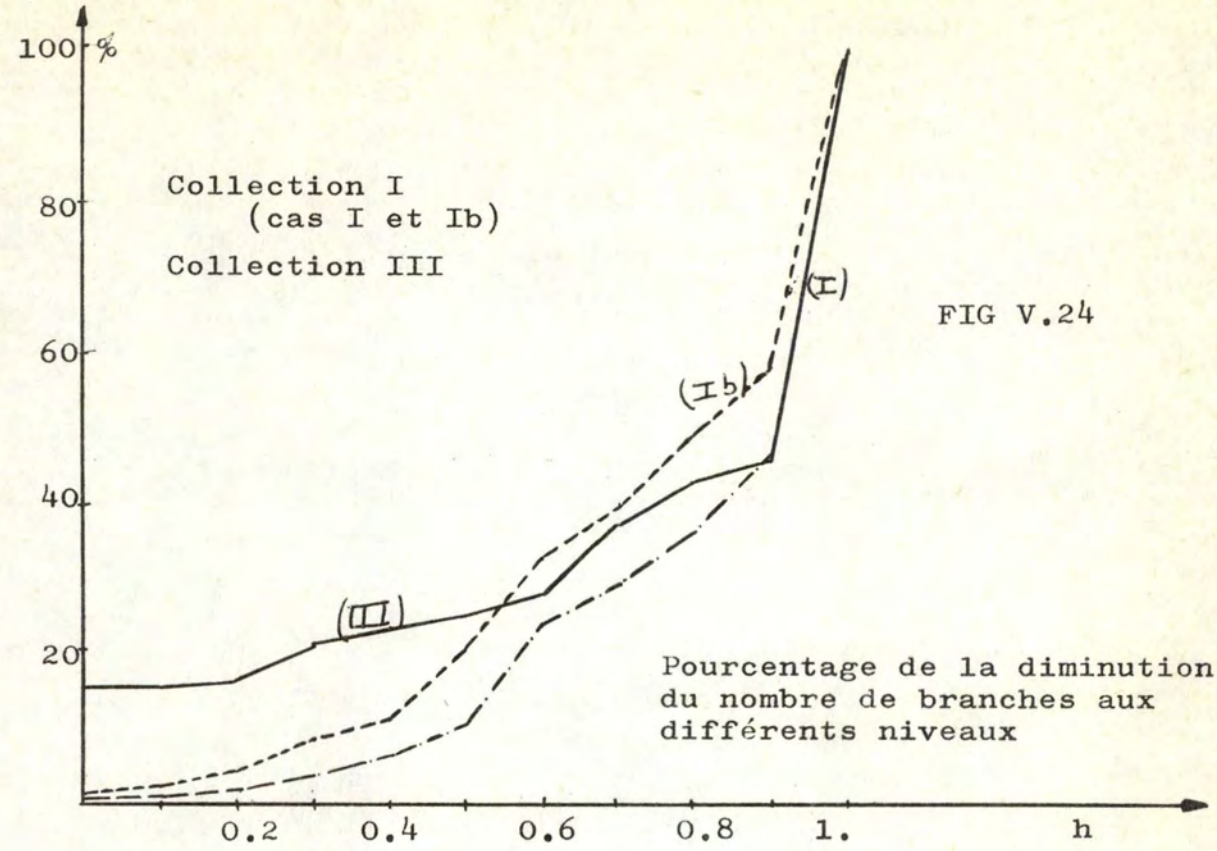


FIG V.24

Pourcentage de la diminution du nombre de branches aux différents niveaux

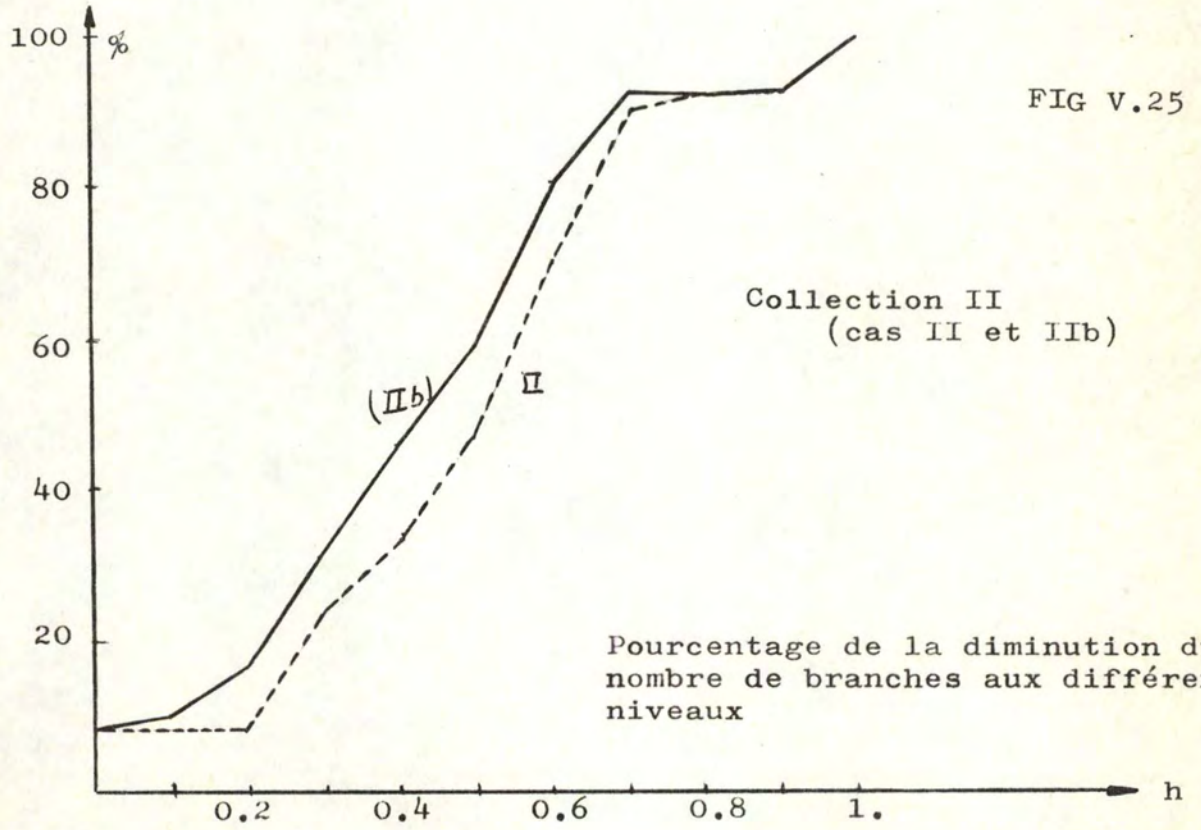


FIG V.25

Collection II
(cas II et IIb)

Pourcentage de la diminution du nombre de branches aux différents niveaux

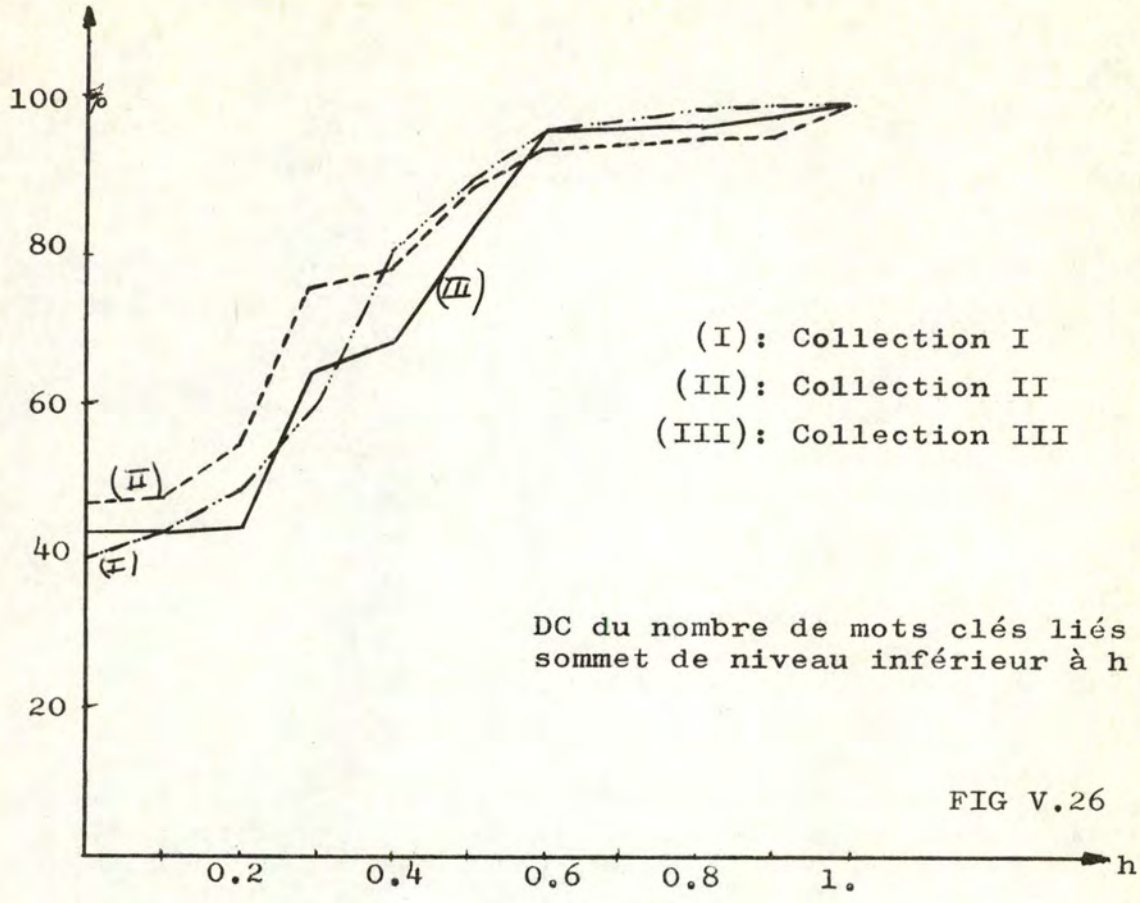


FIG V.26

Ces résultats montrent que la structure de la classification des documents d'une collection est,en général,très différente de celle du vocabulaire de cette collection,bien que l'information de base soit la même(la matrice documents-termes).
Une explication des figures V.21 à V.23 serait la suivante:nous savons qu'un document de la collection I est décrit en moyenne par 7.1 termes contre 3.6 pour II,ce qui permet d'isoler beaucoup plus les documents selon leur contenu pour I que pour II.
Cette hypothèse semble toutefois trop simple pour plusieurs raisons:

- -La collection III,dont les documents sont décrits en moyenne par 4.4 termes(et donc plus proche de II que de I)présente les mêmes caractéristiques que I;
- -Le même phénomène a été observé par [SPARCK-JONES 73] : La collection Cranfield décrite en moyenne par 32 termes par document se prête mieux à l'essaimage que la collection INSPEC décrite en moyenne par 12.2 termes par document.

-Nous avons calculé pour les collections I et III le nombre moyen de termes figurant dans les documents non essaimés:nous obtenons 7.4 pour I et 5.25 pour III.Les documents non essaimés ne sont pas nécessairement ceux décrits par le plus grand nombre de termes.

Nous avons procédé à un réessaimage des documents des collections I et II à partir d'une description modifiée des documents au moyen du thésaurus,c'est à dire que nous avons remplacé chaque terme intervenant dans la description des documents par l'ensemble des termes de la classe correspondante du thésaurus:Les résultats sont exposés sur les figures V.21 à V.25.Le but de cette transformation est double:

-Rendre le vocabulaire moins spécifique(voir tableau V.2)

-Supprimer les différences entre documents dues à l'emploi de mots-clés différents pour décrire le même contenu.

Les courbes(I b)et(II b)des figures V.21 et 22 montrent une amélioration de 10 à 15% entre 0.3 et 0.9 pour I b et entre 0.2 et 0.7 pour II b.Nous voyons qu'il existe toujours une proportion importante de documents non essaimés pour la collection I et l'essaimage des mots clés reste de loin le plus fin.
Il semble donc que les documents diffèrent entre eux par leur contenu plutôt que par l'emploi de descripteurs différents ayant même signification;cette hypothèse devrait toutefois être confirmée par des expériences complémentaires,par exemple,construire d'autres thésaurus,ou encore,supprimer des mots clés dans l'indexation.

V.5 Système questions-réponses

Nous n'avons pas relevé de diagramme Recall-Précision.Il nous aurait fallu préalablement constituer un ensemble de questions-réponses obtenues à partir d'un public,le plus large possible, afin d'avoir une idée représentative des besoins d'information de l'utilisateur.C'est pourquoi,nous nous sommes bornés à étudier l'effet du thésaurus sur quelques cas de matching documents-requête,dans les conditions suivantes:

a Aucun emploi du thésaurus.

b Requête complétée par thésaurus et description initiale des documents

-avec un poids 1 aux termes du thésaurus et 1 aux termes initiaux.

-avec un poids 1 aux termes du thésaurus et 2 aux termes initiaux.

-avec un poids 1 aux termes du thésaurus et 3 aux termes initiaux.

c Requête initiale et description des documents complétée par thésaurus.

d Requête complétée par thésaurus et description complétée par thésaurus des documents(dans les conditions b)

Les collections II et III étant décrites par un vocabulaire très spécifique,nous n'avons pu mettre en évidence l'augmentation de précision.Par contre,on obtient des améliorations parfois très importantes pour le recall.
Nous donnons ci-dessous les différentes questions et les tableaux des valeurs du degré de similarité entre la question et le document sorti par l'algorithme de retrieval.
Dans la question,les nombres entre parenthèses indiquent la fréquence d'emploi du terme dans le vocabulaire d'indexation;la question comprend les termes de l'utilisateur(soulignés)et les termes ajoutés au moyen du thésaurus.
Dans le tableau,les nombres entre parenthèses donnent le classement établi selon les différentes méthodes;la 1ère colonne donne le classement idéal.

Question 1:analyse de la covariance(1), analyse spectrale(3), analyse multivariée(1), série chronologique(3)

Question 2:intervalle de confiance(1),espérance mathématique(1), estimation(2), incertitude(2), probabilité bayésienne(2), analyse séquentielle(1), systèmes dynamiques(1)

Question 3:épreuves d'hypothèse(2), test chi-carré(1), corrélation(1), échantillonnage(1), estimation(2), max.de vraisemblance(1), régression(3).

Question 4:max.de vraisemblance(1), échantillonnage(1), épreuves d'hypothèse(2), moindres carrés(2).

Question 5:espace métrique(6), espace normé(6), espace préhilbertien(1), espaces topologiques(4) espace de Hilbert(7), applications linéaires(2) base(1)

dimension(1), espace dual(1), formes bilinéaires(2), formes canoniques, (2) formes linéaires(1).

Question 6: méthode Adams-Bashforth(2), méthodes d'intégration par pas liés(3), méthodes d'intégration pas séparés(1), méthodes du prédicteur-correcteur(5), consistance(1), méthode d'extrapolation(2), méthode de Runge-Kutta(7)

Question 7: arborescence(1), connexité(1), graphes(3), anneaux(2), dénombrement(2), équivalence(1), graphes(3) lois de composition interne(2), préordre(1).

Question 8: ensemble(5), relations(4), algèbre booléenne(2), anneaux(2), lois de composition interne(2), treillis(3)

Question 9: microprocesseurs(1), microprogrammation(2), adressage(1), architecture, de systèmes(4), protection software(1), système d'interruptions(1).

Question10: processus concurrents(2), gestion du processeur(3), files d'attentes(2), pagination(6), gestion de la mémoire(6), inclusion mutuelle(1), partage de données(1), sémaphore(1)

Question11: théorie des langages(1) analyse syntaxique(2) grammaire formelle(1) analyse de langages(1) cybernétique(1) générateur de compilateur(1) théorie de l'information

Question12: consoles(1) langage Help(1) langage Joss(2) mode conversationnel(1)

Pour les tableaux: signification des colonnes

1ère colonne: classification manuelle

2ème colonne: retrieval sans intervention du thésaurus(requête initiale

3ème colonne: description initiale des documents ;requête complétée au moyen du thésaurus
poids 1 aux termes initiaux,1 aux termes ajoutés

4ème colonne: idem que pour la 3ème mais avec poids 2 pour les termes initiaux

5ème colonne: requête initiale;description des documents complétée par thésaurus

6ème colonne: idem que pour 3 mais avec une description des documents complétée par thésaurus.

Les questions(1),(2),(3),(4)montrent comment on peut améliorer le recall de la chaîne documentaire :une question imprécise au départ, est complétée en incluant dans la requête,les différents sens que prend ce terme dans le thésaurus;on n'a pas en effet,à priori,

Question 1

1	(1) .5	(1) .7	(1) .67	(1) .41	(1) .51
2	(2) .35	(4) .25	(4) .31	(3) .29	(2) .36
2	(2) .35	(2) .50	(2) .47	(2) .35	(3) .44
4	(4) .25	(3) .35	(3) .33	(4) .17	(4) .21
5	-	-	-	-	-

Question 2

1	(1) .33	(1) .88	(1) .84	(5) .24	(1) .55
2	-	(4) .14	(4) .12	(3) .28	(4) .43
3	-	(5) .13	(5) .11	(3) .28	(5) .43

Question 3

1	(1) .53	(1) .71	(1) .73	(2) .64	(2) .73
1	(2) .29	(2) .62	(2) .57	(1) .70	(1) .76
3	-	(3) .13	(3) .10	(3) .25	(4) .27
4	-	(3) .13	(4) .03	(4) .20	(6) .18

Question 4

1	(1) .41	(1) .82	(1) .77	(2) .33	(1) .67
2	-	(3) .19	(3) .14	(3) .28	(3) .42
3	-	-	-	-	-
4	-	(2) .29	(2) .22	(1) .45	(1) .67

Question 7

(1)	(1) .32	(1) .69	(1) .61	(3) .51	(3) .24
(2)	(4) .18	(3) .21	(3) .22	(3) .22	(2) .41

Question 8

1	(3) .43	(1) .49	(2) .52	(3) .33	(2) .58
2	(5) .19	(3) .45	(5) .40	(4) .30	(3) .51
3	(1) .31	(2) .46	(1) .55	(2) .38	(1) .65
4	(4) .31	(4) .44	(3) .44	(5) .27	(4) .47
5	(2)	(5) .29	(4) .41	(1) .53	(5) .31

Question 5

1	(1) .52	(2) .33	(1) .44	(1) .47	(3) .32
2	(2) .45	(3) .29	(3) .38	(2) .45	(4) .29
3	(9) .20	(7) .13	(6) .17	(7) .20	(7) .13
4	(3) .30	(5) .19	(4) .26	(3) .27	(5) .17
5	(8) .22	(6) .14	(5) .19	(6) .21	(6) .14
6	(15) .11	(1) .56	(2) .42	(15) .10	(1) .48

Question 6

1	(1) .38	(3) .43	(2) .43	(4) .38	(3) .43
1	(1) .38	(1) .71	(1) .61	(1) .53	(1) .80
1	(3) .22	(2) .50	(3) .41	(1) .53	(1) .80
4	(4) .21	(4) .24	(4) .24	(5) .19	(4) .21
5	(5) .13	(5) .19	(5) .17	(6) .13	(5) .19
6	(6) .11	(6) .08	(6) .10	(7) .10	(6) .15

Question 11

1	(1) .26	(3) .47	(3) .43
2	(2) .81	(2) .50	(1) .68
3	(3) .23	(1) .58	(2) .50

Question 10

1	(1) 1.	(1) .8	(1) .93
2	(4) .23	(4) .28	(5) .21
3	(6) .12	(2) .38	(4) .28
4	(3) .32	(5) .25	(3) .30
5	(5) .17	-	-
6	(2) .42	(3) .34	(2) .40

Question 9

1	(1) .70	(1) .61	(1) .72
2	(2) .18	(2) .53	(2) .45
3	-	(3) .15	(3) .11
4	-	(4) .14	(4) .10

Question 12

1	(1) .41	(1) .82	(1) .78
2	-	(2) .25	(1) .19

de raison d'écarter telle ou telle signification d'un terme de la requête initiale.D'autre part,en procédant ainsi,la question est formulée dans le cadre du langage d'indexation employé et le couplage documents-requête s'effectue sur davantage de termes.
Les questions (5) et (10),au contraire montrent que la combinaison de plusieurs classes du thésaurus prmet d'améliorer la précision du retrieval.Quel que soit le but poursuivi,le thésaurus demeure toujours un interface entre l'utilisateur et l'indexeur.
Toutefois,selon la fréquence d'un terme et sa survenance avec d'autres termes;il peut exister de profondes divergences entre la portée d'un terme dans le thésaurus construit à partir d'une collection particulière et la signification tacitement admise de ce terme par la majorité des utilisateurs:c'est le cas par exemple pour la question (9):l'intervention du thésaurus déforme complètement la pensée de l'utilisateur;il se fait que dans la collection étudiée,l'on parle(tout à fait accessoirement)de microprogrammation dans le document suivant:
WATSON:Time sharing system and design cncepts.
qui contient le terme microprogrammation comme descripteur.
Comme le terme microprogrammation n'a qu'une fréquence 2,il est étroitement lié aux notions de Time Sharing, allocation de ressource,gestion du processeur,...

V.6 Comparaisons avec d'autres expériences

L'accord avec les résultats publiés par [AUGUSTSON et MINKER 1972] est en général excellent:ils distinguent également 4 zones caractéristiques pour le seuil de signification $\delta < 0.4$; $0.4 \leq \delta \leq 0.5$;$0.5 < \delta < 0.6$; $\delta \geq 0.6$.
Le fait que ces régions ne sont pas les mêmes que dans nos essais, est dû au choix du coéfficient de similarité(AUGUSTSON et MINKER emploient la mesure de TANIMOTO).AUGUSTSON et MINKER ne s'expliquent pas la zone $0.5 < \delta < 0.6$ de grande instabilité,ils ne donnent pas beaucoup d'informations sur les collections employées,mais nous croyons que,comme dans nos expériences,l'allure des courbes dans cette région est due à la rupture des liaisons entre termes de fréquence 1 et termes de fréquence 2:la corrélation entre de tels termes est en effet 0.5.
Il nous est beaucoup moins facile de comparer nos résultats avec SPARCK-JONES et VAN RIJSBERGEN,les méthodes employées et les buts poursuivis étant très différents.D'une part,leur découpage en termes fréquents et non fréquents est beaucoup plus rudimentaire(suivant qu'un terme intervient plus de 20 fois ou non comme descripteur).D'autre part,la description de nos documents est beaucoup plus élémentaire,tant au point de vue complétude de la description qu'au point de vue soin apporté;ainsi nous n'avons pas éprouvé de difficultés pour éliminer les termes fréquents du thésaurus,c'est-à-dire d'empêcher toute possibilité de substitution pour de tels termes.Enfin,SPARCK-JONES examine l'effet de l'emploi d'un thésaurus directement sur les diagrammes Recall-Précision,ce qui nécessite une programmation beaucoup plus complexe et la création au préalable d'un ensemble de questions-réponses construit manuellement.

Nous avons toutefois relevé comme eux la différence entre une classification de documents et une classification de mots clés ainsi que l'existence "d'anomalies" dans les dendrogrammes des documents de certaines collections.
Enfin, nos conclusions quant aux conditions que doit respecter la construction d'un thésaurus pour améliorer le thésaurus sont à peu près les mêmes.

CHAPITRE VI.

CONCLUSIONS ET PROJETS

L'étude fréquentielle de la construction et de l'emploi de thésaurus de termes et de documents apporte plus d'informations que les relevés classiques de taille et nombre de classes d'un thésaurus.
Les conclusions que nous tirons de nos expériences sont étroitement liées au type d'indexation choisi(indexation superficielle et non contrôlée).

L'intervalle de variation du seuil de signification δ peut être divisé en 4 régions.Seule la 2ème dans laquelle la taille moyenne et le nombre de classes sont relativement stables,est utilisable.Dans cette région,le thésaurus n'est plus influencé par le bruit de fond constitué par les liaisons peu significatives entre termes et l'on observe un maximum de possibilités de substitution pour les termes;les classes sont généralement petites: 2,3 ou 4 termes,ce qui convient très bien pour l'extension des requêtes.La manière d'indexation adoptée rend inutile tout traitement particulier pour les termes fréquents,ceux-ci disparaissant rapidement du thésaurus dans la zone de stabilité.
Pour des valeurs croissantes de δ,les classes du thésaurus tendent à être formées de termes qui surviennent un même nombre de fois dans les mêmes documents;la méthode avantage donc les termes rares et très rares au détriment des termes fréquents.
Le mode principal de formation des classes d'un thésaurus est le suivant:à partir d'un noyau de termes de même fréquence,plusieurs classes sont constituées en ajoutant au noyau un terme qui à une fréquence supérieure d'une unité par rapport à celle du noyau; pour les grandes valeurs de δ,il ne subsiste que le noyau.
Une classification de documents diffère considérablement d'une classification de mots,bien que l'information de base soit la même. De plus,les classifications de documents peuvent être très différentes entre elles.Nos expériences semblent indiquer que la forme d'une classification de documents est liée au degré de recouvrement des termes du thésaurus.
Enfin,et c'est une caractéristique essentielle,la description d'un document dépend de tous les autres par les propriétés statistiques du vocabulaire d'indexation.L'indexeur voit chaque document indépendamment de tous les autres,selon son contenu.Au contraire,un système de documentation automatique n'accède au contenu des documents qu'à travers les cooccurrences de termes;et ces cooccurrences sont induites par tous les autres documents.En d'autres termes,le rôle d'un terme appartenant au vocabulaire d'indexation est déterminé,non par la signification tacite que lui attribue la majorité des utilisateurs,mais par l'usage qu'on en fait en tant que descripteur.Une mauvaise indexation peut donc changer totalement la portée des termes employés dans une collection particulière.Nous illustrons cette assertion au moyen de l'exemple déjà abordé au paragraphe V.5.
Soit le livre WATSON :Time Sharing System and design concepts décrit par adressage(1),allocation de ressource(7),architecture de systèmes(4),évaluation de systèmes(4),gestion de fichiers(10),

gestion du processeur(3),gestion de tâches(7),mesure de systèmes(4), microprogrammation(2),pagination(6),protection de software(1), segmentation(3),système d'exploitation(2),système d'interruption (1),Time Sharing(11).
L'indexation a été faite avec beaucoup de soin.Pourtant,ce document est responsable de la formation d'une classe accidentelle constituée par adressage,microprogrammation,protection de software,système d'interruption et contribue également à former la classe microprogrammation et architecture de systèmes.
Dans l'esprit de l'indexeur,le terme microprogrammation n'a été ajouté que pour compléter la description du document en notant l'interférence de la microprogrammation avec les problèmes de Time Sharing:l'intervention de la microprogrammation est accessoire.Au contraire,pour l'algorithme d'essaimage,les termes adressage, microprogrammation,protection du software,système d'interruption constituent les descripteurs les plus importants du document,car ils n'interviennent qu'une ou deux fois dans toute la collection. De plus,les liaisons entre ces termes sont beaucoup plus fortes que celles entre Time Sharing,système d'exploitation,gestion de fichiers et gestion de tâches,ce qui ne correspond pas à notre entendement.En fait l'indexeur a voulu raffiner la description du document,mais il a de ce fait construit une classe accidentelle: le système ne peut savoir,en effet,si le terme ajouté est utilisé à bon escient ou non,comme descripteur principal ou comme complément.
Cet exemple illustre la lacune principale du mode d'indexation adopté:pris individuellement,les documents sont bien indexés,mais globalement,leur description est biaisée,car,l'indexeur a tendance à ne faire figurer dans son indexation que les aspects non évidents,à priori,du contenu des documents,car il estime que les parties évidentes ne permettent pas de distinguer un document d' un autre.Mais ce faisant,les données perçues par le système de documentation automatique peuvent être totalement différentes.Nous donnons ci-dessous d'autres exemples résultant de la politique d' indexation choisie.
C'est ainsi que le mot operating system ne figure pas comme descripteur de documents qui contiennent pourtant le mot operating systems dans leur titre.Par contre,on a employé ce mot pour indexer un livre de programmation,l'auteur traitant de l'influence d'un O.S sur un langage de programmation.
On pourrait de même être étonné que des mots clés tels que intervalle de cônfiance,espérance mathématique,maximum de vraisemblance, échantillonnage,variable aléatoire dans la collection II et codes-objets,mémoire virtuelle,algorithme et organigramme dans la collection III soient des termes rares où très rares.
Une conclusion identique peut être tirée pour les termes processus markoviens et tests statistiques:l'indexeur a jugé inutile d'employer le premier terme pour décrire des documents de files d' attente,mais il l'a par contre noté pour certains livres de théorie des probabilités:de cette manière,il pouvait distinguer entre eux les livres de théorie des probabilité.De même,le terme tests statistiques(2) ne figure dans aucun livre de statistique,mais il a par contre été employé comme descripteur de livres de simulation, l'indexeur ayant voulu souligner l'emploi de tests statistiques pour justifier certaines méthodes de simulation.

En résumé,lorsque l'indexeur décrit un nouveau document,il doit toujours être conscient de l'interaction de ce nouveau document avec tous les autres documents.
Les autres remarques que nous formulons sont beaucoup moins importantes et plus faciles à corriger.Elles ont trait au danger d'employer un vocabulaire trop spécifique,à la complétude et à l'incohérence de la description de documents.
Il y a un avantage certain à employer un vocabulaire très spécifique,car cela permet de limiter,lors d'une requête,le nombre de documents à consulter.Toutefois,le thésaurus étant formé en majorité de classes de fréquence 1 et 2 est trop sensible à des erreurs dans les données d'entrée et à l'ajoute de documents décrits par des mots clés déjà employés.
La complétude d'une description de documents est importante si l'on désire des performances très élevées d'une chîne documentaire: on ne peut pas,par exemple,espérer un recall important pour la collection II où un document n'est décrit,en moyenne,que par 3.6 termes.
Enfin,nous avons noté certaines incohérences dans la description des documents:certains d'entre eux,à contenu identique,étaient indexés de différentes façons;c'est ainsi que les livres de PL/I ont été indexés par langage PL/I,langage de programmation,programmation. Pour l'un d'entre eux,on y a ajouté, sans raison valable, les principaux mécanismes de la programmation PL/I(boucle DO, instructions de rupture de séquence,traitement de liste...)

De l'étude que nous avons entreprise,il semble qu'une bonne indexation de documents devrait comporter:

- Des termes à portée générale,connus par la majorité des utilisateurs,pour lesquels les possibilités de substitution seront très réduites.
- Des termes à fréquence modérée,qui permettent de localiser le domaine d'intérêt.
- Des termes de fréquence faible qui permettent de distinguer entre eux des petits groupes de documents:ce sont ces termes que l'utilisateur ne connaît généralement pas:mais il suffit qu'il en connaisse un ou deux pour compléter la question au moyen du thésaurus.

Les expériences que nous avons réalisées ne nous permettent cependant pas de tirer des règles d'indexation précises et valables pour n'importe quelle indexation.Les directives que nous formulons devraient donc être vérifiées au moyen d'expériences complémentaires(voir projets).

1 L'indexeur ne peut ignorer le but final de son travail,c'est-à-dire:
   - la construction automatique de classifications non hiérarchiques à partir du relevé statistique des co-occurrences de mots clés.
   - la consultation automatique de ces classes(au moyen de stratégies de recherche)afin de reformuler,dans le langage d'indexation,une requête initiale incomplète ou imprécise.

2 Une description doit être complète même si certains descripteurs apparaissent,aux yeux de l'indexeur,superflus parce qu'

évidents ou même redondants;en particulier,le choix des descripteurs ne doit pas être guidé par des relations de synonimie ou d'hiérarchie entre descripteurs.De plus,l'indexation doit être logique:il n'omettra un descripteur pour décrire un document,que si peu d'utilisateurs s'intéressent à ce seul aspect du document,ou encore,si ce descripteur ne figure pas dans l'indexation d'autres documents avec une signification particulière ou dans un contexte inhabituel.Enfin,un descripteur inutile ou évident pour un indexeur(opérant sur un document concret) peut ne plus l'être pour la majorité des utilisateurs.

3 L'indexeur ne retiendra pas dans sa description des détails manifestement secondaires:il est inutile de tenir compte des subtilités de style de chaque auteur ou de distinguer entre eux des documents différents à contenu identique.Dans l'optique thésaurus,une description fine risque d'ailleurs de donner une mauvaise représentation du contenu du document.

4 L'indexeur doit être conscient de l'usage qui sera fait du document:pour un article d'une revue spécialisée,on demandera une description précise;au contraire,pour des livres,on préfèrera plutôt une indexation permettant l'accès à plusieurs catégories d'utilisateurs selon les différents sujets abordés. Les remarques (2) et (3) s'appliquent tout particulièrement à l'indexation de livres.

5 En cas d'hésitation,l'indexeur pourra utilement consulter les parties du thésaurus de documents et du thésaurus de mots clés concernées par le nouveau document,afin de maintenir l'indexation cohérente.

PROJETS

Il semble naturel de prévoir que les techniques exposées dans ce mémoire ne seront réellement utiles que si elles permettent la construction dynamique et la consultation automatique de thésaurus comportant des milliers de documents et de termes.
Trois problèmes sont à envisager:

a Il faut disposer de procédures de mises à jour simples pour les documents et les termes qui sont ajoutés progressivement aux thésaurus existants, sans trop dégrader les performances de retrieval.

b Après un nombre suffisamment important de mises à jour, il est nécessaire de procéder à un réessaimage de tous les objets: le mieux est de pouvoir se servir de l'essaimage déjà effectué.

c L'extension à de grosses collections de documents nécessite des procédures d'essaimage plus économiques que la plupart de celles proposées à l'heure actuelle (qui font intervenir le carré du nombre d'objets à essaimer).

Des solutions ont été proposées par [JOHNSON et LAFUENTE], [KERCHNER], [LESSER], [BRAUEN] et appliquées à une même collection de documents. Il serait intéressant d'appliquer ces techniques ou des variantes à d'autres collections afin de dégager des critères généraux, indépendantes des caractéristiques propres d'une collection.
Nous pensons néanmoins avant d'envisager des applications pratiques, qu'il est plus important de pouvoir prévoir qualitativement et quantitativement le comportement d'une collection et d'une indexation de documents, lorsqu'elle est incluse dans un système de documentation automatique utilisant des thésaurus. On s'est aperçu, en effet, que des collections de documents, indexés avec le même soin, pouvaient donner lieu à des performances de retrieval très différentes [SPARCK-JONES 73]. Il faudrait donc dégager des critères simples permettant, préalablement à l'essaimage de toute la collection, d'évaluer l'influence de l'homogénéité, du type d'indexation adopté, de la technique d'essaimage, de la formulation des requêtes et de la stratégie de retrieval sur les performances globales de la chaîne documentaire. Nous pensons à des expériences où l'on ferait varier:

- le nombre de documents de chaque collection
- l'indexation, en ajoutant ou retranchant-sélectivement ou non-des descripteurs

Ces 2 expériences doivent permettre d'évaluer la sensibilité du thésaurus

- la formulation des questions selon que l'on connaît ou non la composition du thésaurus.
- l'influence des différents termes du vocabulaire par des techniques de pondération. Il importe que cette pondération soit faite automatiquement d'après l'usage des termes, une pondération établie par l'indexeur comportant beaucoup trop d'arbitraire. Comme cette pondération doit imiter notre jugement, elle n'est envisageable que dans le cas d'indexations très soignées et complètes.

```
      PROGRAM  ASTER
C**** PROGRAM ASTER = PROGRAMME DE CONSTRUCTION DES ESSAIMS
C  ** IL APPELLE LES ROUTINES SUIVANTES :
C**     -RWDATA:ENVOI DE LA MATRICE DOC-TERMES SUR DISQUE (LIGNE PAR LIGNE ET
C**                                 COLONNE PAR COLONNE
C**     -ITITKW:ECRITURE SUR DISQUE DES TITRES DE DOCUMENTS ET DES MOTS  CLES
C**     -CLIMAX:CONSTRUVTION DES CLIQUES MAXIMALES
C**     -CONNEX: CONSTRUCTION DES COMPOSANTES CONNEXES
C**     -COSINE:CONSTRUCTION DE LA MATRICE TERMES-TERMES ET CONSTRUCTION DE LA
C                        DE LA TABLE DE BITS PERMETTANT D'ADRESSER LA MATRICE.
C**** PARAMETRES NECESSAIRES POUR L'EXECUTION
C**     -NBTERM :INDIQUE LE NOMBRE DE TERMES
C**     -NBDOC : INDIQUE LE NOMBRE DE DOCUMENTS
C**     -NFICH1--NFICH9 INDIQUENT LES NUMEROS DE FICHIERS UTILISES (FICHIERS
C                  PERMANENTS ET FICHIERS DE MANOEUVRE  )
C      **NFICH1 : FICHIER CONTENANT LA MATRICE DOC.TERMES COL PAR COL
C      ** NFICH2 : FICHIER CONTENANT LES MOTS CLES
C      ** NFICH3 : FICHIER CONTENANT LA MATRICE TERMES TERMES SANS CUTOFF
C      ** NFICH4 : FICHIER CONTENANT LES TITRES DE DOCUMENTS
C      ** NFICH5 : FICHIER CONTENANT LA MAT  DOC.TERMES LIGNE PAR LIGNE
C      ** NFICH6 : FICHIER CONTENANT LE THESAURUS
C      ** NFICH8 : FICHIER AVEC LES QUESTIONS COMPLETEES PAR LE THESAURUS
C      ** NFICH9 : FICHIER MEMORISANT L'ARBORESCENCE
C**     CUTOFF : INDIQUE LA VALEUR DU SEUIL DE SIGNIFICATION
C**    IPTCOS,IDATA,ITITKW,ICONN,NDISK SONT DES SWITCHES SPECIFIES A L'EXECUTION
C**       IPTCOS INDIQUE S'IL FAUT CALCULER LA MATRICE TERMES - TERMES OU
C                        S'IL FAUT CALCULER LA TABLE DE BITS COPRESPONDANT AU
C                        SEUIL DE SIGNIFICATION ADOPTE
C**       -IDATA INDIQUE S'IL FAUT LIRE ET STOCKER LA MATRICE DOC-TERMES OU NON
C**       -ICLIM SPECIFIE SI ON DEMANDE LES CLIQUES MAXIMALES OU NON
C         -ICONN  IDEM QUE ICLIM,MAISC POUR LES COMPOSANTES CONNEXES
C**       -NDISK PERMET D'EXIGER LA MEMORISATION DU THESAURUS SUR DISQUE
C*** ZONES DE MEMOIRES PRINCIPALES
C       -LOGIC(250,128) POUR LA TABLE DE BITS (PASSEE PAR COMMON  )
C       - MOTCLE : POUR GARDER LES MOTS CLES EN MEMOIRE CENTRALE (PASSEE PAR
C                                                                 COMMON )
C       - UNE ZONE AUXILIAIRE (BIDON) NECESSAIRE DANS LES ROUTINES CLIMAX ET
C                                                                 CONNEX
      LOGICAL*1 BIT(1)
      LOGICAL*1 LOGIC(1000,128),MOTCLE(5000),CARTE(80)
      INTEGER*2 BIDON(10000)
      INTEGER*4 BITEST
      COMMON/ZONE/LOGIC
      COMMON/ZONAUX/BIDON
      COMMON/KEY/MOTCLE
      EQUIVALENCE (BIT(1),LOGIC(1))
      READ 5,NBTERM,NBDOC
      READ5,NFICH1,NFICH2,NFICH3,NFICH4,NFICH5,NFICH6,NFICH7,NFICH8
 5    FORMAT(10I5)
      READ 6,CUTOFF
 6    FORMAT(F5.3)
      LCLE= NBTERM*20
      READ 60,(MOTCLE(L),L=1,LCLE)
 60   FORMAT(80A1)
      PRINT 61,(MOTCLE(L),L=1,LCLE  )
 61   FORMAT(T10,20A1,5X,20A1,5X,20A1,5X,20A1     ,5X,20A1)
      CALL TITKW(NBTERM,NBDOC,NFICH5,NFICH2,NFICH4,ITITKW    )
      CALL CHRONO(1)
      CALL RWDATA(NBTERM,NBDOC,NFICH5,NFICH1,IDATA)
      CALL CHRONO(2)
 666  CONTINUE
      CALL CHRONO(1)
      CALL COSINE (NBTERM,NBDOC,CUTOFF,NFICH1,NFICH3,IPTCOS)
      CALL CONNEX(NBTERM,2,NFICH3,NFICH4,NFICH6,ICONN   )
      CALL CLIMAX(NBTERM,NDISK,NFICH3,NFICH4,NFICH6,NO,TOTES,ICLIM  )
      CALL CHRONO(2)
      READ(5,236,END=560) CUTOFF,IPTCOS,NDISK,ICLIM,ICONN
 236  FORMAT(F5.2,8I5)
      GO TO 666
 560  CONTINUE
      STOP
      END
```

```
      SUBROUTINE CLIMAX(NBTERM,ISW,NFICH3,NFICH4,NFICH6,NO,TOTES,ICLIM)
C******* BUT DE CLIMAX: CONSTRUIRE LES CLIQUES MAXIMALES
C    *** PARAMETRES TRANSMIS
C     * NBTERM NOMBRE D'OBJETS A ESSAIMER (TERMES OU DOCUMENTS )
C     * ISW SWITCH SPECIFIANT LES DIFFERENTES OPTIONS
C           =1  ON ECRIT LE THESAURUS DE TERMES SUR DISQUE ET ON L'IMPRIME
C           =2 ON IMPRIME LE THESAURUS
C           =6 AUCUNE MEMORISATION
C     * NFICH3 INDIQUE LE FICHIER OU SE TROUVE STOCKEE LA MATRICE TERMES-TERMES
C       NFICH4 SPECIFIE LE FICHIER AVEC LES TITRES DE DOCUMENTS
C       NFICH6 SPECIFIE LE FICHIERN SUR LEQUEL EST ECRIT LE THESAURUS
C       NO  RENVOIE AU PROGRAMME PRINCIPAL LE NOMBRE TOTAL D'ESSAIMS DE TAILLE
C                                                   SUPERIEURE A 1
C       TOTES  DONNE LE NOMBRE TOTAL D'ESSAIMS
C    *** ZONE MEMOIRE  UTILISEE
C     * BIT : TABLE DE BITS TRANSMISE PAR COMMON
C     * MOTCLE : ZONE DE STOCKAGE DES MOTS CLES
C     * STACK  VECTEUR QUI CONTIENT LES SOMMETS DE LA CLIQUE EN COURS DE
C                                                         CONSTRUCTION
C     * CANDI VECTEUR QUI CONTIENT LES CANDIDATS
C                  J EST CANDIDAT SI CANDI(J) = 1 SINON CANDI(J) = 0
C     * TAKEOF  VECTEUR MEMORISANT LES SOMMETS QUI NE SONT PLUS CANDIDATS
C                           PAR SUITE DE LA SELECTION D'UN SOMMET
C     * MEMOR  VECTEUR MEMORISANT LA HAUTEUR DE LA PILE TAKEOF A CHAQUE
C                           SELECTION D'UN CANDIDAT
C   ** ROLE DES PRINCIPALES VARIABLES
C     * ADTOCA (=ADD TO CANDIDATES ) EST TRANSMIS A LA ROUTINE UPDATE LORSQUE
C               L'ON SUPPRIME  LE DERNIER SOMMET DE LA CLIQUE ET QUE L'ON REND A
C               < CANDIDATES > LES SOMMETS ENLEVES PRECEDEMMENT PAR LA
C               SELECTION DE CE SOMMET
C     *SUTOCA (=SUBTRACT TO CANDIDATES ) EST TRANSMIS A LA ROUTINE UPDATE
C               LORSQU'ON ENLEVE DES SOMMETS A < CANDIDATES > SUITE A LA
C               SELECTION D'UN SOMMET
C     * JCAN INDIQUE LE SOMMET COURANT QUE L'ON ANALYSE
C     * ALPHA , BETA  VOIR SIGNIFICATION DANS LE MEMOIRE
C     * K MEMORISE LA HAUTEUR COURANTE DE STACK
C     * JTAKE  IDEM POUR TAKEOF
C     * KMEMO  IDEM POUR MEMOR
C     * BOOL VOIR SIGNIFICATION DANS MEMOIRE
C   ** ROLE DES ROUTINES UPDATE ,GAMMA , ENUMER , ISOLE
C     * UPDATE FAIT LA MISE A JOUR DE CANDI,TAKEOF MEMOR ACHAQUE MODIFICATION DE
C               LA CLIQUE(AJOUTE OU SUPPRESSION D'UN SOMMET )
C     * GAMMA EST UNE FUNCTION DE VALEUR 1 OU 0 SUIVANT QUE LA CLIQUE
C                CONSTRUITE EST MAXIMALE OU NON
C     * ISOLE ROUTINE PARCOURANT LA TABLE DE BITS POUR DETECTER LES SOMMETS ISOLES
C     * ENUMER  ROUTINE PARAMETREE PAR LE SWITCH ISW : SON BUT EST DE MEMORISER L
C               LA CLIQUE MAXIMALE CONSTRUITE
      LOGICAL*1 BIT(1),MOTCLE(1)
      INTEGER*2 STACK(1000),CANDI(1000),MEMOR(1000),TAKEOF(1000),
     1 BILAN(1000)
      INTEGER*4 TOT,SUM,SOM
      COMMON/ZONE/BIT
      COMMON/ZONAUX/STACK,CANDI,MEMOR,TAKEOF,BILAN
      COMMON/KEY/MOTCLE
      INTEGER*4 TOTES
      INTEGER*4 ADTOCA,SUTOCA,NBTERM,ALPHA,BETA,JTAKE,K,KMEMO  , GAMMA
      LOGICAL*1 BOOL
      GO TO (1000,2000),ICLIM
1000  PRINT 1
    1 FORMAT(1HO/1H ,T50,'ENONCE DES CLIQUES MAXIMALES'//)
      DO 2 I=1,5000
2     STACK(I) = 0
      NO=0
      ADTOCA=1
      SUTOCA=0
      DO 20 JCAN=1,NBTERM
   20 CANDI(JCAN) = 1
      I=1
   38 JTAKE = 0
      ALPHA = I
      BOOL = .TRUE.
      BETA = I
      K=1
      STACK(K) = I
```

```
      CALL UPDATE(SUTOCA,NBTERM,K,JTAKE)
      JCAN=STACK(K)+1
   33 KMEMO=K
   24 IF(CANDI(JCAN)-1) 22,23,22
   23 K=K+1
      STACK(K) = JCAN
      CALL UPDATE(SUTOCA,NBTERM,K,JTAKE)
   22 JCAN = JCAN+1
      IF(JCAN- NBTERM) 24,24,25
   25 IF(K-KMEMO) 26,26,27
   27 IF(ALPHA-BETA) 28,30,28
   28 IF(GAMMA(CANDI,NBTERM,ALPHA,BETA)-1) 30,31,30
   30 IF(GAMMA(CANDI,NBTERM,1,I-1)-1) 32,31,32
   31 JCAN = STACK(K) + 1
      CALL UPDATE(ADTOCA,NBTERM,K,JTAKE)
      K=K-1
      GO TO 33
   32 NO = NO + 1
      BILAN(K) = BILAN(K)+1
      CALL ENUMER(MOTCLE,STACK,K,NO,NBTERM,ISW,NFICH3,NFICH4,NFICH6)
   26 IF(K-2) 34,35,35
   35 BETA = STACK(K)
      JCAN= BETA+1
      CALL UPDATE(ADTOCA,NBTERM,K,JTAKE)
      K=K-1
      IF(.NOT.BOOL) GO TO 41
      ALPHA = STACK(K)
      BOOL = .FALSE.
      GO TO 33
   41 IF(STACK(K) - ALPHA) 42,33,33
   42 ALPHA = STACK(K)
      GO TO 33
   34 I=I+1
      IF(I-NBTERM+1) 36,36,37
36    CALL UPDATE(ADTOCA,NBTERM,K,JTAKE)
      GO TO 38
37    CALL ISOLE(NBTERM,NBALON,ISW,NFICH3,NFICH4,NFICH6,NO+1)
      BILAN(1) = NBALON
      TOTES = NO + NBALON
      PRINT 102
  102 FORMAT(1H0/1H0,T10,'RESUME D''INFORMATIONS GLOBALES SUR LES TECHNI
     1QUES D''ESSAIMAGE'//)
      PRINT 207
  207 FORMAT(T35,'COL NO 1 TAILLE , COL NO 2 NBRE')
      K=1
  304 IF(BILAN(K) ) 301,301,302
  302 PRINT 303,K,BILAN(K)
  303 FORMAT(T35,I10,5(2H *),I4)
      TOT=TOT+K*BILAN(K)
      SUM=SUM + BILAN(K)
  301 K=K+1
      IF(K-NBTERM) 304,304,305
  305 TAILLE = FLOAT(TOT) / FLOAT(SUM)
      PRINT 199,TAILLE
  199 FORMAT(1H0,T30,'TAILLE MOYENNE DES ESSAIMS',F12.4)
2000  RETURN
      END
```

```
      SUBROUTINE CONNEX(NBSOM,ISW,NFICH3,NFICH4,NFICH6,ICONN  )
C       * CONNEX * BUT CONSTRUIRE LES COMPOSANTES CONNEXES
C          SOMCAN MEMORISE LES SOMMETS RESTANTS PARMI LESQUELS ON CHERCHE LES
C          COMPOSANTES CONNEXES
C         INITIALEMENT TOUS LES SOMMETS SONT CANDIDATS
C          STATE EST UN VECTEUR FIXANT L'ETAT DES SOMMETS
C          LORSQU'ON EXAMINE UN SOMMET I, TOUS LES SOMMETS J ADJACENTS A I SONT
C          MARQUES STATE(I) = 1
C                LORSQUE TOUS LES SOMMETS ADJACENTS DE I ONT ETE MARQUES ON MAR-
C               QUE  STATE(I) = 2
C          L'ALGORITHME PROCEDE DONC COMME SUIT
C          TOUS LES SOMMETS  CANDIDATS SE TROUVENT DANS SOMCAN
C          1) PRENDRE LE 1ER SOMMET DE SOMCAN ET MARQUER STATE(J) = 2
C          2) MARQUER 1 TOUT SOMMET ADJACENT
```

```
C           3) REPRENDRE TOUS LES SOMMETS K MARQUES 1 ETPOUR CHAQUE SOMMET K,
C              MARQUER STATE(K) = 2
C            MARQUER STATE(L) = 1 TOUT SOMMET L ADJACENT A K ET
C                   MARQUÉ INITIALEMENT STATE(L) = 0
C           IL Y A AUTANT D'ITERATIONS QUE DE COMPOSANTES CONNEXES.
C          * VARIABLES AUXILIAIRES *
C           NBCAN INDIQUE LE NOMBRE DE SOMMETS RESTANTS PARMI LESQUELS ON RECHER-
C              CHE  LES C.C.
C           NBCA VARIABLE AUXILIAIRE POUR L'EVALUATION DE NBCAN
C           NBCOM COMPTE LE NOMBRE DE SOMMETS DANS UNE C.C.
C           NOCOM  VECTEUR MEMORISANT SOMMETS D'UNE C.C. AFIN DE POUVOIR TRANSMET-
C              TRE LA C. C. A ENUMER
C          BILAN  VECTEUR SERVANT A RESUMER LA TAILLE ET LE NOMBRE DE CLIQUES
C           NBSOM NOMBRE DE SOMMETS (=NOMBRE D'OBJETS A ESSAIMER)
      INTEGER*2 BIDON(100)
      INTEGER*4 TOT,SUM,SOM,NBTERM
      LOGICAL*1 BIT(1)
      INTEGER*2 STATE(1000),SOMCAN(1000),NOCOM(1000),BILAN(1000)
      LOGICAL*1 BACK,MOTCLE(1)
      INTEGER*4 NBSOM,NBCOM,NBCAN
      INTEGER*4 BITEST
      COMMON/ZONE/BIT
      COMMON/ZONAUX/STATE,SOMCAN,NOCOM,BILAN
      COMMON/KEY/MOTCLE
      DO 400 I=1,NBTERM
      JJ=0
      DO 500 J=1,NBTERM
         IF(BITEST(I,J,BIT)-1)500,600,600
600   JJ=JJ+1
      BIDON(JJ)=J
500   CONTINUE
      PRINT 300,I,JJ,(BIDON(L),L=1,JJ)
300   FORMAT(1H0,2I5,(T20,20I4))
400   CONTINUE
      GO TO (1000,2000),ICONN
1000  DO 249 I=1,NBSOM
      SOMCAN(I)=I
249   BILAN(I) = 0
      PRINT 744
744   FORMAT(1H1,T30,'IMPRESSION DES COMPOSANTES CONNEXES'///)
      NBCOMP=0
      NBCAN=NBSOM
131   IF(NBCAN-2) 1,2,3
2     I=SOMCAN(1)
      J=SOMCAN(2)
      IF(BITEST(I,J,BIT) -1) 1,5,1
5     STATE(J)=2
1     STATE(SOMCAN(1))=2
      GO TO 15
3     BACK=.FALSE.
      KI=1
14    I=SOMCAN(KI)
      STATE(I)=2
      KJ=2
10    IF(KJ-KI) 6,7,6
6     J=SOMCAN(KJ)
      IF(STATE(J)-1)  8,7,7
8     IF(BITEST(I,J,BIT)-1) 7,9,7
9     BACK=.TRUE.
      STATE(J)=1
7     KJ=KJ+1
      IF(KJ-NBCAN) 10,10,11
11    KI=KI+1
      IF(KI-NBCAN) 12,12,13
12    IF(STATE(SOMCAN(KI))-1) 11,14,11
13    IF(.NOT.BACK) GO TO 15
      BACK=.FALSE.
      KI=2
      GO TO 12
15    NBCA=0
      NBCOM=0
      KI=1
21    I=SOMCAN(KI)
      IF(STATE(I)-1) 17,18,19
17    NBCA=NBCA+1
```

```
      SOMCAN(NBCA) = I
      GO TO 20
   19 NBCOM=NBCOM+1
      NOCOM(NBCOM) = I
      STATE(I) = 0
   20 KI=KI+1
      IF(KI-NBCAN) 21,21,22
   22 NBCAN=NBCA
      NBCOMP= NBCOMP +1
      CALL ENUMER(MOTCLE,NOCOM,NBCOM,NBCOMP,NBSOM,ISW,NFICH3,NFICH4,
     1 NFICH6)
      BILAN(NBCOM) = BILAN(NBCOM) + 1
      IF(NBCAN) 132,132,131
  132 CONTINUE
      PRINT 102
  102 FORMAT(1HJ/1HO,T10,'RESUME D''INFORMATIONS GLOBALES SUR LES TECHNI
     1QUES D''ESSAIMAGE'//)
      PRINT 207
  207 FORMAT(T35,'COL NO 1 TAILLE , COL NO 2 NBRE')
      K=1
  304 IF(BILAN(K) ) 301,301,302
  302 PRINT 303,K,BILAN(K)
  303 FORMAT(T35,I10,5(2H *),I4)
      TOT=TOT+K*BILAN(K)
      SUM=SUM + BILAN(K)
  301 K=K+1
      IF(K-NBSOM) 304,304,305
  305 TAILLE = FLOAT(TOT) / FLOAT(SUM)
      PRINT 199,TAILLE
  199 FORMAT(1HO,T30,'TAILLE MOYENNE DES ESSAIMS',F12.4)
      RETURN
  999 FORMAT(1HO,'ERREUR DANS CONNEX')
   18 PRINT 999
 2000 RETURN
      END

      SUBROUTINE UPDATE(OFFON,NBTERM,K,JTAKE)
C***** BUT DE UPDATE  MISE A JOUR DES ZONES CANDI,TAKEOF,MEMOR,LORS DE LA [illegible]I
C               MODIFICATION DE LA COMPOSITION D'UNE CLIQUE (MAX OU MIN )
C           BITEST(I,J,BIT) EST UNE ROUTINE ASSEMBLER RENVOYANT LA VALEUR 1 OU 0
C               SUIVANT QUE L'ARÊTE I,J DU GRAPHE EXISTE OU NON
C           OFFON=SUTOCA SI ON RETRANCHE DES CANDIDATS AU VECTEUR < CANDI >
C                 =ADTOCA SI ON REND DES CANDIDATS ENLEVES ANTERIEUREMENT A
C                                  < CANDI >
      LOGICAL*1 BIT(1)
      INTEGER*2 STACK(1000),CANDI(1000),MEMOR(1000),TAKEOF(1000),
     1 BILAN(1000)
      COMMON/ZONE/BIT
      COMMON/ZONAUX/STACK,CANDI,MEMOR,TAKEOF,BILAN
      INTEGER*4 BITEST
      INTEGER*4 OFFON,NBTERM,K,SOM,ELEM,L,JTAKE,ADTOCA,SUTOCA
      ADTOCA=1
      SUTOCA=0
      IF(OFFON-SUTOCA) 2,1,2
    1 SOM=0
      ELEM=STACK(K)
      L=1
    6 IF(CANDI(L)-1) 3,4,3
4   IF(BITEST(ELEM,L,BIT)-1) 5,3,5
    5 CANDI(L) =0
      SOM = SOM+1
      JTAKE = JTAKE + 1
      TAKEOF(JTAKE) = L
    3 L=L+1
      IF(L-NBTERM) 6,6,7
    7 MEMOR(K) = SOM
    9 RETURN
    2 SOM = MEMOR(K)
   10 IF(SOM) 9,9,8
    8 SOM= SOM-1
      CANDI(TAKEOF(JTAKE))=1
      JTAKE= JTAKE-1
      GO TO 10
      END
```

```
      SUBROUTINE ISOLE(NBTERM,NBALON,ISW,NFICH3,NFICH4,NFICH6,NOCLUM)
C     * ISOLE * ENUMERE LES ESSAIMS REDUITS A UN SEUL ELEMENT
C           CHAQUE LIGNE DE LA MATRICE EST PARCOURUE
C           LE FONCTIONNEMENT EST EXPLIQUE AU PARAGRAPHE II.3.1 DU
C                2 CAS PEUVENT SE PRESENTER
C                  - L'ON TROUVE UN ELEMENT NON NUL : PASSAGE A LA LIGNE SUIVANTE
C                  - TOUS LES ELEMENTS SONT NULS: LA LIGNE PARCOURUE CORRESPOND A
C                      UN ESSAIM D'UN SEUL TERME
      LOGICAL*1 BIT(1)
      INTEGER*2 SLONE(1000),CANDI(1000),MEMOR(1000),TAKEOF(1000),BILAN(1000)
     1000)
      INTEGER*2 SALONE(1000)
      INTEGER*4 BITEST
      COMMON/ZONE/BIT
      COMMON/ZONAUX/SALONE,CANDI,MEMOR,TAKEOF,BILAN
      COMMON/KEY/MOTCLE
      LOGICAL*1 MOTCLE(1)
      INTEGER*4 NBALON
      NBALON=0
      I=1
    5 J=1
3     IF(BITEST(I,J,BIT)) 1,1,2
    1 J=J+1
      IF(J-NBTERM) 3,3,4
    4 NBALON=NBALON + 1
      SALONE(NBALON) = I
    2 I=I+1
      IF(I-NBTERM) 5,5,6
    6 IF(NBALON) 7,7,8
    8 PRINT 100
  100 FORMAT(1HO,T2,'ENSEMBLE DES MOTS CLES ISOLES')
      CALL ENUMER(MOTCLE,SALONE,NBALON,NOCLUM,NBTERM,ISW,
     1 NFICH3,NFICH4,NFICH6   )
      DO 60 J=1,NBTERM
60    CANDI(J) =0
      I=1
13    CANDI(SALONE(I)) = NOCLUM - 1 + I
      I=I+1
      IF(I-NBALON) 13,13,14
14    ID=NOCLUM
      WRITE(NFICH6'ID)(CANDI(J),J=1,NBTERM  )
    7 RETURN
      END

      INTEGER FUNCTION GAMMA(CANDI,NBTERM,FIRST,SECOND)
C****** BUT DE GAMMA  DIRE SI LA CLIQUE EST MAXIMALE OU NON
C                    POUR CE TEST , ON PARCOURT UNE PORTION DE CANDI
C                    POUR CHAQUE CLIQUE SUSCEPTIBLE D'ETRE MAXIMALE,CETTE ROUTINE
C                          EST APPELEE 2 FOIS
C       ** ROLE DES PARAMETRES
C            * CANDI VECTEUR INDIQUANT LES SOMMETS CANDIDATS
C            * NBTERM INDIQUE LE NOMBRE D'OBJETS A ESSAIMER
C            * FIRST , SECOND : INDIQUENT LES LIMITES INFERIEURE ET SUPERIEURE
C                    DES NUMEROS DE SOMMETS ENTRE LESQUELS NE PEUT FIGURER UN
C                    SOMMET CANDIDAT SI LA CLIQUE EST MAXIMALE
C$$$$$* BUT DE ISOLE  IMPRIMER LES CLIQUES MAXIMALES REDUITES A UN SEUL ELEMENT
C                    ET LES ECRIRE SUR DISQUE
C                    LA TABLE DE BITS EST TRANSMISE PAR COMMON
      INTEGER*2 CANDI(250)
      INTEGER*4 NBTERM,FIRST,SECOND,M
      M=FIRST
    5 IF(M-SECOND) 1,1,2
    1 IF(CANDI(M)-1) 3,4,3
    3 M=M+1
      GO TO 5
    4 GAMMA=1
      RETURN
    2 GAMMA=0
      RETURN
      END
```

```
1   BYTBIT   START
2   *    * BYTBIT * CONSTRUIT UNE TABLE DE BITS
3   *             LA MATRICE TRIANGULAIRE INFERIEURE EST STOCKEE LIGNE PAR LIGNE,
4   *                CHAQUE ELEMENT N'OCCUPANT QU'UN SEUL BIT
5   *             CETTE ROUTINE EST APPELEE PAR COSINE
6   *             1ER ARGUMENT  : L'ADRESSE DU VECTEUR IAUX(INTEGER*2)      IAUX
7   *                      CONTIENT LA LIGNE NO I
8   *             2EME ARGUMENT  : DONNE LA VALEUR DE I
9   *             LA ROUTINE COMPORTE 2 BOUCLES
10  *                LA 1ERE BOUCLE EST EXECUTEE LORSQUE 32 BITS ONT ETE FORMES DANS E
11  *             LE REGISTRE 9 : ON STORE ALORS CE REGISTRE A L'ADRESSE SPECIFIEE
12  *                      PAR R5
13  *             LA 2EME BOUCLE CONSTRUIT LE CONTENU DU REGISTRE  9 : A CHAQUE
14  *             STEP LE CONTENU DU REGISTRE 9 EST DECALE D'UNE POSITION VERS
15  *                LA GAUCHE
16            STM   14,12,12(13)
17            BALR  3,0
18            USING *,3
19            L     4,0(1)
20            L     5,4(1)
21            L     6,8(1)
22            L     6,0(6)
23            LA    10,1
24            LA    11,2
25            XR    8,8
26            XR    9,9
27            LA    15,31
28            NR    15,6
29            SRL   6,5(0)
30            LTR   15,15
31            BZ    SUITE
32            AR    6,10
33  SUITE     LH    8,0(4)

34            ALR   9,8
35            AR    4,11
36  BOUCLE2   L     7,=F'32'
37  BOUCLE1   LH    8,0(4)
38            SLL   9,1(0)
39            ALR   9,8
40            AR    4,11
41            BCT   7,BOUCLE1
42            ST    9,0(5)
43            LA    5,4(5)
44            BCT   6,BOUCLE2
45            LM    14,12,12(13)
46            BR    14
                . .
47

48            END
```



```
1   BITEST   START
2   *    * BITEST * TESTER S'IL EXISTE UNE ARETE RELIANT 2 SOMMETS QUELCONQUES
3   *             DU GRAPHE : POUR CELA ON TESTE SI UN BIT EST A 1 OU 0
4   *          3 ARGUMENTS SONT TRANSMIS
5   *
6   *                I ET J SONT LES SOMMETS ETUDIES
7   *             BIT DONNE L'ADRESSE DE LA ZONE DE STOCKAGE
8   *             R3 CONTIENT LA VALEUR DE I
9   *             R5 CONTIENT LA VALEUR DE J
10  *                SI I=J ALORS RETURN
11  *             SI I SUPERIEUR A J ALLER EN NEXT
12  *             SI I INFERIEUR A J PERMUTER I ET J (LE GRAPHE N'EST PAS ORIENTE)
13  *             ON CONSTRUIT LA FORMULE
14  *                   ADD(G(I,J)) = ADD(BIT) + (I-1)*(LONGUEUR LIGNE) + J-1
```

```
15              STM   14,12,12(13)
16              BALR  2,0
17              USING *,2
18              LA    10,1
19              L     3,4(1)
20              L     3,0(3)
21              L     5,0(1)
22              L     5,0(5)
23              CR    5,3
24              BH    NEXT
25              BE    NOSUCCES
26              LR    6,5
27              LR    5,3
28              LR    3,6
29  NEXT        L     6,=F'7'
30              NR    6,3
31  *                   R6 CONTIENT LE RESTE DE LE DIVISION DE J PAR 8 : ON SAIT A
32  *                   PRESENT LE NO DU BIT A TESTER
33              SR    5,10
34              M     4,LENGTH
35  *                         R4 CONTIENT (I-1) LONGUEUR D'UNE LIGNE

36              SRL   3,3(0)
37  *                   R3 CONTIENT LE QUOTIENT ENTIER DE J PAR 8
38              AR    5,3
39              A     5,8(1)
40  *                      R3 CONTIENT L'ADRESSE DU BYTE A TESTER
41  *                   ON CONSTRUIT UNE TABLE DE BRANCHEMENTS POUR TESTER          DES
42  *                    BITS DE 0 A 7
43  *                     LES DIFFERENTS ELEMENTS DE LA TABLE DE BRANCHEMENTS CONTIENNENT
44  *                      LE TEST UNDER MASK ADEQUAT
45  *                    LES 2 BRANCHEMENTS POSSIBLES (SUCCES OU INSUCCES )
46              SLL   6,4(0)
47              B     TABLE+0(6)
48  TABLE       SR    5,10
49              TM    0(5),X'01'
50              BZ    NOSUCCES
51              B     FIN
52              DS    CL2
53  FIRST       TM    0(5),X'80'
54              BZ    NOSUCCES
55              B     FIN

56              DS    CL4
57  SECOND      TM    0(5),X'40'
58              BZ    NOSUCCES
59              B     FIN

60              DS    CL4
61  THIRD       TM    0(5),X'20'
62              BZ    NOSUCCES
63              B     FIN

64              DS    CL4
65  FOURTH      TM    0(5),X'10'
66              BZ    NOSUCCES
67              B     FIN

68              DS    CL4
69  FIFTH       TM    0(5),X'08'
70              BZ    NOSUCCES
71              B     FIN

72              DS    CL4
73  SIXTH       TM    0(5),X'04'
74              BZ    NOSUCCES
75              B     FIN

76              DS    CL4
77  SEVENTH     TM    0(5),X'02'
78              BZ    NOSUCCES
79              B     FIN

80              DS    CL4
```

```
81   NOSUCCES XR    10,10
82   *               POSITIONNEMENT DU REGISTRE 0 SAUVE DANS LA ZONE AUXILIAIRE
83   FIN      ST    10,20(13)
84            LM    14,12,12(13)
85            BR    14
86   LENGTH   DC    F'128'
87
88            END
```

```
A FORTRAN IV (VER L38) SOURCE LISTING:  RWDATA  SUBROUTINE     05/13/75     PAGE  0008

    1        SUBROUTINE RWDATA(NBTERM,NBDOC,NFICH1,NFICH2,IDATA)
    2 C     * RWDATA * ENREGISTRER LA MATRICE DOCUMENTS TERMES
    3 C           LIGNE PAR LIGNE SUR NFICH5 ET COLONNE PAR COLONNE SUR NFICH1
    4 C            LES DONNEES SONT ENTREES LIGNE PAR LIGNE ,PAR GROUPE DE 100
    5 C           SAUF EVENTUELLEMENT POUR LE DERNIER GROUPE,LA FIN DE FICHIER ETANT
    6 C           DETECTEE PAR UN NOMBRE <0 DANS ZONE(I5) DE LA CARTE D'ENTREE
    7 C            LES LIGNES SONT DECODEES:
    8 C               G(I,J) = 1    SI LE JEME TERME INTERVIENT DANS LE DOC NO I
    9 C              LEUR STOCKAGE SUR NFICH5 EST IMMEDIAT
   10 C           LA CONSTRUCTION DES COLONNES DEMANDE PLUS DE PRECAUTIONS:
   11 C           ON NE PEUT ECRIRE UNE PORTION D'ENREGISTREMENT: IL FAUT DONC LIRE
   12 C           CHAQUE ENREGISTREMENT,MODIFIER LA PARTIE DESIREE ET
   13 C             REECRIRE L'ENREGISTREMENT
   14 C           LES ENREGISTREMENTS SONT SOUS FORME D'INTEGER*2
   15 C               NFICH1 FICHIER CONTENANT LES LIGNES DE LA MATRICE
   16 C               NFICH2 FICHIER CONTENANT LES COLONNES DE LA MATRICE
   17        LOGICAL*1 G(100,1000),LAUX(2000),BOOL(2),TRU,FAL
   18        INTEGER*2 JOJO,IAUX(1000),ICOL(30)
   19        EQUIVALENCE (JOJO,BOOL(1)),(BOOL(1),FAL),(BOOL(2),TRU),
   20       1 (LAUX(1),IAUX(1))
   21        COMMON/ZONE/G
   22        GO TO (47,48),IDATA
   23 47     JOJO=1
   24        JOJO=1
   25        DO 111 I=1,250
   26 111    WRITE(NFICH2'I)(IAUX(L),L=1,250)
   27        ASSIGN 19 TO IBACK
   28        DO 50 I=1,30
   29 50     ICOL(I)=0
   30        ICRT=0
   31        DO 49 I=1,1000
   32 49     IAUX(I)=0
   33        IX=0
   34        LOWLIG=0
   35        LOWCOL=0
   36                  DO 7 I=1,100
   37        DO 7 J=1,NBTERM
   38 7      G(I,J)=FAL
   39 56     READ 51,INUM,(ICOL(K),K=1,25)
   40 51     FORMAT(I5,25I3)
   41        IF(INUM) 52,52,53
   42 53     IL=INUM-IX*100
   43        IF(IL-100) 54,54,55
   44 54     K=1
   45 58     J=ICOL(K)
   46        IF(J) 56,56,57
   47 57     G(IL,J) = TRU
   48        ICRT=ICRT+1
   49        K=K+1
   50        GO TO 58
   51 55     DO 10 I=1,100
   52        DO 11 J=1,NBTERM
   53 11     LAUX(J+J) = G(I,J)
   54        ID=LOWLIG+I
   55 10     WRITE(NFICH1'ID) (IAUX(L),L=1,NBTERM)
   56        DO 15 J=1,NBTERM
   57        READ(NFICH2'J)(IAUX(L),L=1,NBDOC)
   58        DO 16 I=1,100
   59 16     LAUX(LOWCOL+I+I) = G(I,J)
```

```
60 15     WRITE(NFICH2'J)(IAUX(L),L=1,NBDOC)
61        GO TO IBACK,(19,20)
62 19     LOWLIG=LOWLIG+100
63        LOWCOL=LOWCOL+200
64        IX=IX+1
65        DO 13 I=1,100
66        DO 13 J=1,NBTERM
67 13      G(I,J)=FAL
68           GO TO 53
69 52     IF(LOWLIG-900) 60,60,61
70 60     ASSIGN 20 TO IBACK
71        GO TO 55
72 61     PRINT 63
73 63     FORMAT(1HO,5(1H*),'TOO MUCH DOCUMENTS(MORE THAN 1000')
74        STOP
75 20     PROCEN = FLOAT(ICRT)*100./FLOAT(NBTERM*NBDOC)
76        PRINT 5,NBDOC,NBTERM,PROCEN
77 5      FORMAT(1HO/1HO/1HO,T5,125(1H*)/T5,1H*,
78       1T40,'NOMBRE DE DOCUMENTS',I5,T124,1H*/T5,1H*,
79       2T40,'NOMBRE DE MOTS CLES',I5,T124,1H*/T5,1H*,
80       3T40,'DENSITE DE LA MATRICE DOCUMENTS-TERMES',F8.3,T124,1H*/
81       4T5,125(1H*))
82 48     RETURN
83        END

  1         SUBROUTINE COSINE(NBTERM,NBDOC,CUTOFF,NOFICH,NFICH3,IPTCOS)
  2 C     * COSINE  EST DIVISEE EN 2 PARTIES
  3 C         →DANS LA PARTIE I ,LA MATRICE TRIANGULAIRE INFERIEURE DE SIMILARITE EST
  4 C              CONNSTRUITE LIGNE PER LIGNE, PUIS ENVOYEE SUR DISQUE
  5 C            LA MESURE DE SIMILARITE EMPLOYEE EST LA CORRELATION COSINUSOIDALE
  6 C            FIRST ET SECOND SONT 2 VECTEURS DE TRAVAIL QUI LORS DU CALCUL DE
  7 C              S(I,J) CONTIENNENT LES COLONNES I ET J DE LA MATRICE RECTANGULAIRES
  8 C               DOCUMENTS - TERMES
  9 C            IAUX CONTIENT LA IEME LIGNE DE LA MATRICE DE SIMILARITE
 10 C              (LA VALEUR DE S(I,J) EST MULTIPIEE PAR 10000 ET CONVERTIE EN ENTIER
 11 C         →DANS LA 2EME PARTIE , ON MET LA MATRICE DE SIMILARITE SOUS FORME D'UNE
 12 C               TABLE DE BITS POUR UNE VALEUR DONNEE DE CUTOFF
 13 C            UN SWITCH IPTCOS PERMET DE FAIRE LES PARTIES I ET II OU
 14 C              UNIQUEMENT LA PARTIE II
 15        INTEGER*4 SOMFIR,SOMSEC,INTER
 16        LOGICAL*1 BIT(1)
 17        INTEGER*2 FIRST(1000),SECOND(1000),IAUX(1000)
 18        INTEGER*4 T1,T2
 19        COMMON/ZONE/BIT
 20        COMMON/ZONAUX/FIRST,SECOND,IAUX
 21        DO 499 I=1,3000
 22 499    FIRST(I)=0
 23        ICRT=0
 24        GO TO (129,130),IPTCOS
 25 129    T1=1
 26 2      READ(NOFICH'T1)(FIRST(K),K=1,NBDOC)
 27        T2=1
 28 3      IF(T2-T1) 4,5,5
 29 4      INTER=0
 30        SOMFIR=0
 31        SOMSEC=0
 32        READ(NOFICH'T2)(SECOND(K),K=1,NBDOC)
 33        DO 21 I=1,NBDOC
 34        IF(FIRST(I)-1) 22,23,22
 35 23     SOMFIR = SOMFIR+1
 36        IF(SECOND(I)-1) 21,25,21
 37 25     INTER=INTER+1
 38 27     SOMSEC = SOMSEC + 1
 39        GO TO 21
 40 22     IF(SECOND(I) - 1) 21,27,21
 41 21     CONTINUE
 42        IF((SOMFIR.EQ.0).OR.(SOMSEC.EQ.0)) GO TO 28
 43        CORARI = INTER/SQRT(FLOAT(SOMFIR*SOMSEC))
 44        GO TO 39
 45 28     CORARI=0.
 46 39     IAUX(T2) = CORARI*10000
 47        T2=T2+1
 48        GO TO 3
 49 5      IAUX(T2) = 0
 50        WRITE(NFICH3'T1) (IAUX(L),L=1,T1)
```

```
      PRINT 883,T1
  883 FORMAT(I10)
      T1=T1+1
      IF(T1-NBTERM) 2,2,30
   30 CONTINUE
  130 IVAL=CUTOFF*10000
      IBIT=1
      DO 500 I=1,NBTERM
      READ(NFICH3'I) (IAUX(L),L=1,I)
      DO 501 J=1,I
      IF(IAUX(J) -IVAL) 502,503,503
  502 IAUX(J) = 0
      GO TO 501
  503 IAUX(J) = 1
      ICRT=ICRT+1
  501 CONTINUE
      CALL BYTBIT(IAUX,BIT(IBIT),I)
      IBIT=IBIT+128
  500 CONTINUE
      PROCEN=FLOAT(ICRT)*100./FLOAT(NBTERM*NBTERM)
      PRINT 6,NBTERM,CUTOFF,PROCEN
    6 FORMAT(1H0/1H0/1H0,T5,125(1H*)/T5,1H*,
     1T40,'NOMBRE DE TERMES',I5,T124,1H*/T5,1H*,
     2T40,'VALEUR DE CUTOFF POUR LA CONSTRUCTION DU 1-GRAPHE',F6.3,T124,
     3  1H*,/T5,1H*,
     4T40,'DENSITE DE LA MATRICE TERMES-TERMES',F8.3,T124,1H*/
     5T5,125(1H*)///)
      RETURN
      END

      SUBROUTINE TITKW(NBTERM,NBDOC,NFICH1,NFICH2,NFICH4,ITITKW)
C     * TITKW * BUT ECRIRE LES TITRES DE DOCUMENTS ET LES MOTS CLES
C        UN SWITCH PERMET DE DEMANDER L'UN DES TRAVAUX SUIVANTS
C          1) ENREGISTREMENT DESC MOTS CLES SUR NFICH2 ET IMPRESSION DES
C                                                          MOTS CLES
C          2) ENREGISTREMENT DES TITRES DE DOCUMENTS SUR NFICH4
C          3) IMPRESSION DES TITRES DE DOCUMENTS ET DES MOTS CLES QUI LES DECRI-
C            VENT
      LOGICAL*1 MOTCLE(5000),CARTE(80)
      INTEGER*4 LOWER,UPPER
      INTEGER*2 BIDON(2000),IVEC(2000)
      COMMON/KEY/MOTCLE
      COMMON/ZONAUX/BIDON,IVEC
      GO TO (1000,2000,2000),ITITKW
 1000 DO 10 ID=1,NBDOC
      READ 11,(CARTE(L),L=1,80)
   11 FORMAT(80A1)
   10 WRITE(NFICH4'ID)(CARTE(L),L=1,80)
      GO TO 3000
      LCLE=NBTERM*20
      READ 12,(MOTCLE(L),L=1,LCLE)
   12 FORMAT(60A1)
      ID=0
      LOWER=1
      UPPER=1000
    4 ID=ID+1
      WRITE(NFICH2'ID)(MOTCLE(L),L=LOWER,UPPER)
      LOWER=LOWER+1000
      UPPER=UPPER+1000
      IF(UPPER-LCLE-1000) 4,4,3000
 2000 ID=0
      LOWER=1
      UPPER=1000
   14 ID=ID+1
      READ(NFICH2'ID)(MOTCLE(L),L=LOWER,UPPER)
      LOWER=LOWER+1000
      UPPER=UPPER+1000
      IF(UPPER-LCLE-1000) 14,14,15
   15 IF(ITITKW-2) 85,3000,85
 3000 DO 39 ID=1,NBDOC
      READ (NFICH4'ID)  (CARTE(L),L=1,80)
      PRINT 138,(CARTE(L),L=1,80)
```

```
 43 138   FORMAT(1HO,T10,80A1)
 44       READ (NFICH1'ID)(BIDON(L),L=1,NBTERM)
 45       JJ=0
 46       DO   42 K=1,NBTERM
 47       IF(EIDON(K)) 42,42,43
 48 43    JJ=JJ+1
 49       IVEC(JJ) = K
 50 42    CONTINUE
 51       CALL ENUMER(MOTCLE,IVEC,JJ,ID,NBTERM,2,NFICH3,NFICH4,NFICH6 )
 52 39    CONTINUE
 53 85    RETURN
 54       END

  1       SUBROUTINE ENUMER(MOTCLE,CLUMP,NBMOT,NOCLUM,NBTERM,ISW,NFICH3,
  2      1 NFICH4,NFICH6)
  3 CCCCCC
  4 C
  5 C  ISW=1  ENVOI DES  ESSAIMS DE BASE SUR DISK & IMPRESSION DES ESSAIMS
  6 C   ISW=2  IMPRESSION DES ESSAIMS
  7 C  ISW=3  ENVOI DES ESSAIMS FUSIONNES SUR DISQUE & IMPREESION DES ESSAIMS
  8 C  ISW=4  IMPRESSION DES ESSAIMS FUSIONNES
  9 C  ISW=5  ENVOI DES ESSAIMS DE DOCUM. SUR DISQUE & IMPRESSION DES DOCUMENTS D'UN
 10 C  ISW=6  IMPRESSION DES DOCUMENTS DE CHAQUE ESSAIM
 11       INTEGER*2 CLUMP(250),IAUX(250),IVEC(250)
 12       INTEGER*4 CPTOUT
 13       LOGICAL*1 MOTCLE(1),PRIN(200)
 14       INTEGER*4 NBMOT,NOCLUM,NBTERM,NBDOC,ISW,N1,N2,N3,N4
 15       PRINT 999,NOCLUM
 16  999  FORMAT(1HO,'NO',I5)
 17       GO TO (1,2,3,4,5,6),ISW
 18    1  DO 25 I=1,NBTERM
 19   25  IAUX(I) =0
 20       I=1
 21   26  IAUX(CLUMP(I))=1
 22       I=I+1
 23       IF(I-NBMOT) 26,26,27
 24   27  ID=NOCLUM
 25       WRITE(NFICH6'ID)(IAUX(I),I=1,NBTERM)
 26    2  K=0
 27       IF(NBMOT) 22,22,23
 28   23  CPTOUT=0
 29   67  K=K+1
 30       J=(CLUMP(K)-1)*20
 31       DO 24 I=1,20
 32   24  PRIN(CPTOUT+I) = MOTCLE(J+I)
 33       CPTOUT=CPTOUT+20
 34       IF(K-NBMOT) 101,111,111
 35  111  ASSIGN 22 TO JOB
 36       GO TO 13
 37  101  IF(CPTOUT-100) 67,28,28
 38   28  ASSIGN 23 TO JOB
 39   13  PRINT 100,(PRIN(KI),KI=1,CPTOUT)
 40  100  FORMAT(T2,20A1,5X,20A1,5X,20A1,5X,20A1,5X,20A1)
 41       GO TO JOB,(22,23)
 42   22  RETURN
 43    3  DO 90 I=1,NBTERM
 44   90  IAUX(I) = 0
 45       I=1
 46   32  ID=CLUMP(I)
 47       READ(NFICH6'ID)(IVEC(I),I=1,NBTERM)
 48       DO 31 I=1,NBTERM
 49   31  IAUX(I) = IAUX(I) + IVEC(I)
 50       I=I+1
 51       IF(I-NBMOT) 32,32,33
 52   33  ID= NOCLUM
 53       WRITE(NFICH3'ID)(IAUX(I),I=1,NBTERM)
 54    4  I=1
 55       NBMOT=0
 56   42  IF(IAUX(I)) 40,41,40
 57   40  NBMOT=NBMOT+1
 58       CLUMP(K) = I
 59   41  I=I+1
 60       IF(I-NBTERM) 42,42,2
 61    5  RETURN
 62    6  I=1
```

```
63        PRINT 65,(CLUMP(L),L=1,NBMOT)
64     65 FORMAT(T10,'CLUMP IN ENUMER',3013)
65     64 ID=CLUMP(I)
66        READ(NFICH4'ID) (PRIN(L),L=1,80)
67        PRINT 52,(PRIN(L),L=1,80)
68     52 FORMAT(T10,80A1)
69        I=I+1
70        IF(I-NBMOT) 64,64,53
71     53 RETURN
72        END
```

```
 1        PROGRAM SEARCH
 2 C         BUT DU PROGRAMME SEARCH : APPELER LES DIFFERENTES SOUS-ROUTINES
 3 C
 4 C   * SLINK * CONSTRUIT LE DENDROGRAMME SOUS FORME DE LA REPRESENTATION PAR
 5 C                              POINTEURS
 6 C   * EXLINK * CONSTRUIT L'ARBORESCENCE A PARTIR DU DENDROGRAMME
 7 C   * VECNOD * CONSTRUIT LES VECREURS REPRESENTANT LES NOEUDS ET LES STOCKE SUR
 8 C                              LE FICHIER NFICH5
 9 C   * DEQUES * INTERPRETE LA QUESTION RTBAPELLE LA ROUTINE BSTRA
10 C                              (ROUTINE DE RETRIEVAL )
11 C       LES ROUTINES SONT ASSEMBLEES COMME SUIT
12 C   ( SLINK,SLINK1,SLINK2,SLINK3,DENDR,COSLNK ) CONSTRUCTION ET IMPRESSION DE
13 C                                 DENDROGRAMME
14 C   ( EXLINK,MINUS) : CONSTRUCTION DE L'ARBORESCENCE A PARTIR DU DENDROGRAMME
15 C   (VECNOD) : CONSTRUCTION DES VECTEURS REPRESENTANT LES NOEUDS INTERMEDIAIRES
16 C   (DEQUES,EXQUES,BSTRA,ENUMER,FGSQ,LISDOC ) : CONSULTATION DU THESAURUS DE
17 C                              DOCUMENTS ET IMPRESSION DES RESULTATS
18 C   (CHRONO ) : ROUTINE COMPTABILISANT LE TEMPS CPU ENTRE 2 POINTS DESIRES
19 C       LES VECTEURS NA,HA,NB,HB,ZONA,ZONB SONT NECESSAIRES POUR LA CONSTRUC-
20 C                              TION ET L'IMPRESSION DU DENDROGRAMME
21 C         LES VECTEURS LAMBDA,PISOM,BRO,SON,NEWNAM,NUM, INTERVIENNENT DANS LA
22 C                              CONSTRUCTION DE L'ARBORESCENCE
23 C         TOUS CES VECTEURS SONT DES INTEGER*2
24 C            LE VECTEUR MOT CLE MEMORISE LES MOTS CLES
25 C          NOBJ MEMORISE LE NOMBRE D'OBJETS A ESSAIMER
26 C                           = NBDOC SI CE SONT DES DOCUMENTS
27 C                           = NBTERM SI CE SONT DES MOTS CLES
28 C          RACINE MEMORISE LE NO DU NOEUD CORRESPONDANT A LA RACINE DE
29 C                                 L'ARBORESCENCE
30 C          LE DENDROGRAMME ET L'ARBORESCENCE SONT MEMORISES DANS LE FICHIER NFICH9
31 C                              DU COEFFICIENT DE DISSIMARITE(EST UTILISE DANS LA
32 C                                    DETECTION D'ERREURS )
33        INTEGER*4 TOP
34        INTEGER*4 SIZE,SCALE
35        INTEGER*4 RACINE
36        INTEGER*2 ZONA(1000),ZONB(1000),PISOM(1000)
37        INTEGER*2 NA(1000),HA(1000),NB(1000),HB(1000)
38        INTEGER*2 BRO(4000),SON(4000),NEWNAM(4000),NUM(1000),LAMBDA(1000)
39        EQUIVALENCE(NA(1),PISOM(1)),(LAMBDA(1),HA(1))
40        LOGICAL*1 MOTCLE(5000)
41        EQUIVALENCE(NB(1),NUM(1)),(NEWNAM(1),HB(1))
42        LOGICAL*1 CARTE(80)
43        COMMON/KEY/MOTCLE
44         CALL CHRONO(1)
45        READ 1,NBTERM,NBDOC,NBQUES
46        READ 1,NFICH1,NFICH2,NFICH3,NFICH4,NFICH5,NFICH6,NFICH7,NFICH8,
47       1 NFICH9
48 1      FORMAT(10I5)
49        READ 2, X , Y
50 2      FORMAT(10F5.2)
51         CALL CHRONO(2)
52        TOP=25
53        LCLE=NBTERM*20
54        READ 81,(MOTCLE(L),L=1,LCLE)
55 81     FORMAT(60A1)
56        CALL CHRONO(1)
57        NA(1)=1
58        HA(1)=TOP
59         CALL CHRONO(1)
60        CALL SLINK(NA,NB,HA,HB,TOP,MOTCLE,NBDOC,ZONA,ZONB,NFICH5  )
61        CALL CHRONO(2)
62        CALL CHRONO(1)
63        CALL EXLINK(BRO,SON,LAMBDA,PISOM,NEWNAM,NUM,NBDOC,LENGTH,RACINE )
```

```
64        CALL CHRONO(2)
65           PRINT 453,(BRO(K),K=1,LENGTH)
66           PRINT 453,(SON(K),K=1,LENGTH)
67 453       FORMAT(25I4)
68         CALL CHRONO(1)
69        CALL VECNOD(BRO,SON,NBDOC,LENGTH,NBTERM,NFICH5)
70         CALL CHRONO(2)
71        ID=5
72        WRITE(NFICH9'ID)(BRO(K),K=1,RACINE)
73        ID=6
74        WRITE(NFICH9'ID)(SON(K),K=1,RACINE)
75        ID=7
76        WRITE(NFICH9'ID)  RACINE
77        ID=1
78        WRITE(NFICH9'ID) (NA(K),K=1,NBDOC)
79        ID=2
80        WRITE(NFICH9'ID) (HA(K),K=1,NBDOC)
81        ID=3
82        WRITE(NFICH9'ID)(ZONA(K),K=1,NBDOC)
83        ID=4
84        WRITE(NFICH9'ID)(ZONB(K),K=1,NBDOC)
85        CALL DEQUES(BRO,SON,NBQUES,RACINE,MOTCLE,NBTERM,X,Y,NFICH3,NFICH4,
86       1 NFICH6,NFICH5    )
87 630    READ(5,1,END=560) NPARAM
88           CALL EXQUES(BRO,SON,NBQUES,RACINE,MOTCLE,NBTERM,X,Y,
89       1 NFICH3,NFICH4,NFICH6,NFICH5,46,NPARAM    )
90        GO TO 630
91 999       CONTINUE
92 560    CONTINUE
93        STOP
94        END

 1         SUBROUTINE SLINK(NA,NB,HA,HB,TOP,MOTCLE,NOBJ,NBTERM,ZONA,ZONB,
 2        1 NFICH5  )
 3 C     * SLINK * CONSTRUIT LA REPRESENTATION PAR POINTEURS
 4 C        NA ET HA SONT LES VECTEURS PI ET LAMBDA : A LA IEME ITERATION
 5 C                                NA ET HA MEMORISENT LE DENDROGRAMME FORME PAR LES
 6 C                                I PREMIERS OBJETS
 7 C             A L'ITERATION I SLINK APPELLE COSLNK QUI CALCULE LA IEME LIGNE DE LA
 8 C                                LIGNE DE LA MATRICE DE DISSIMILARITE
 9 C        ZONA,ZONB,SONT 2 VECTEURS SERVANT AU SAUVETAGE MOMENTANE DE NA ET HA
10 C       * SLINK2 * EST APPELE POUR DONNER UNE REPRESENTATION COMMODE DU
11 C                                DENDROGRAMME
12 C             ZONA MEMORISE DANS L'ORDRE LES OBJETS DU DENDROGRAMME
13 C             ZONB INDIQUE LA HAUTEUR CORRESONDANTE POUR CHAQUE OBJET
14 C        SCALE EST UTILISE POUR CALCULER LA DISTORSION IMPOSEE PAR L'ULTRAME-
15 C                                TRIQUE
16 C        NMISS MEMORISE LE NOMBRE D'ERREURS DANS LA MATRICE DE DISSIMILARITE
17 C                                (VALEURS NEGATIVES OU SUPERIEURES A  TOP)
18 C       * DENDR * EST APPELE POUR L'IMPRESSION DU DENDROGRAMME
19        INTEGER*2 ZONA(1),ZONB(1)
20        LOGICAL*1 MOTCLE(2000)
21        INTEGER*2 NA(1),NB(1)
22        INTEGER*2 HA(1),HB(1)
23        INTEGER*4 TOP
24        INTEGER*4 SIZE,SCALE
25        NMISS=0
26        SIZE=0
27        NA(1)=1
28        HA(1) = TOP
29 C
30 C
31        DO 1 I=2,NOBJ
32        I1=I-1
33        NA(I) =I
34        HA(I)=TOP
35        CALL COSLNK(ZONA,ZONB,HB,NBTERM,I,NFICH5    )
36        DO 2 J=1,I1
37        IF(HB(J)) 3,2,2
38 3      HB(J) = TOP-1
39 2      SIZE = SIZE+HB(J)
40        CALL SLINK1(NA,HA,HB,I1,NOBJ)
41 1      CONTINUE
42        DO 13 L=1,NOBJ
```

```
      ZONA(L) = NA(L)
13    ZONB(L) = HA(L)
      CALL SLINK2(ZONA,NB,ZONB,HB,NOBJ)
      PRINT 764,(ZONA(L),L=1,NBTERM)
      PRINT 765,(ZONB(L),L=1,NBTERM)
764   FORMAT(1HO/1HO,T3,'HA',(T10,20I5))
765   FORMAT(1HO/1HO,T3,'NA',(T10,20I5))
      SCALE=0
      DO 5 I=1,NOBJ
      IF(ZONB(I) - TOP +1) 6,5,5
6     IBITU = ZONB(I)
      SCALE=MAXO(SCALE,IBITU)
5     CONTINUE
      CALL DENDR(NOBJ,ZONA,ZONB,0,SCALE,101,MOTCLE,2)
      IF(NMISS) 999,8,9
9     NPAIR = (NOBJ*(NOBJ-1))/2
      PRINT 9008,NMISS,NPAIR
9008  FORMAT(/,1HO,I6,'DC VALUES MISSING FROM ',I6)
      GO TO 999
8     DELHAT = (FLOAT(SIZE)-SLINK3(ZONA,ZONB,NOBJ))/FLOAT(SIZE)
      PRINT 9009,DELHAT
9009  FORMAT(1HO/1HO,'** DISTORSION IMPOSEE PAR L''ULTRAMETRIQUE **',
     1  E12.4)
999   RETURN
      END

      SUBROUTINE SLINK2(NA,NB,HA,HB,NOBJ)
C     * SLINK2 * ROUTINE AUXILIAIRE QUI DONNE UNE REPRESENTATION
C          EQUIVALENTE DU DENDRIOGRAMME  PERMETTANT UNE
C            REPRESENTATION GRAPHIQUE(VOIR SIBSON 73 )
      INTEGER*2 NA(1),NB(1)
      INTEGER*2 HA(1),HB(1)
      INTEGER*4 H
      NB(NOBJ) = NOBJ
      DO 1 N=2,NOBJ
      H= HA(NOBJ+1-N)
      NEXT=NOBJ
2     NOW = NEXT
      NEXT = NB(NOW)
      IF(H-HA(NEXT)) 3,2,2
3     NB(NOW) = NOBJ +1 - N
1     NB(NOBJ+1-N) = NEXT
      NEXT = NB(NOBJ)
4     NOW = NEXT
      N= NA(NOW)
      IF(NB(NOW)) 5,999,6
6     NEXT = NB(NOW)
      IF(NB(N)) 7,999,8
8     NB(NOW) = NB(N)
      NB(N) = -NOW
      NMISS = NMISS + 1
      IF(NEXT-NOBJ) 4,11,999
7     NOSE = -NB(N)
      NB(NOW) = NB(NOSE)
      NB(N) = -NOW
      NA(NOW) = NOSE
      IF(NEXT-NOBJ) 4,11,999
5     NOSE = -NB(NOW)
      NEXT = NB(NOSE)
      IF(NB(N)) 9,999,10
10    NB(NOSE) = NB(N)
      NB(N) = -NOSE
      IF(NEXT-NOBJ) 4,11,999
9     NA(NOW) = -NB(N)
      NB(N)=-NOSE
      N=NA(NOW)
      NB(NOSE) = NB(N)
      IF(NEXT-NOBJ) 4,11,999
11    NEXT = -NB(NOBJ)
      DO 12 N=1,NOBJ
      HB(N)= HA(N)
      NB(N) = NEXT
12    NEXT = NA(NEXT)
      DO 13 N=1,NOBJ
      NOW=NB(N)
```

```
 51 13     HA(N) = HB(NOW)
 52        DO 14 N=1,NOBJ
 53        NOW = NA(N)
 54 14     NB(NOW) = N
 55        RETURN
 56 999    STOP
 57        END
```

```
  1        SUBROUTINE SLINK1(NA,HA,HB,I1,NOBJ)
  2 C    * SLINK1 * EST APPELE A L'ITERATION I PAR SLINK
  3 C           MEMOIRE
  4 C           NA ET HA CONSTITUENT LA REPRESENTATION PI ET MAMBDA
  5 C           HB VECTEUR DE DONNEES = LA LIGNE NO I DES DC
  6 C           I1 = I-1 (L'ELEMENT (I,I) DE LA MATRICE DES DC EST INUTILE )
  7        INTEGER*2 NA(1)
  8        INTEGER*2 HB(1),HA(1)
  9        INTEGER*4 H
 10        DO 1 J=1,I1
 11        NEXT = NA(J)
 12        IF(HA(J)-HB(J)) 2,3,3
 13 2      H= HB(J)
 14        IF(HB(NEXT)-H) 1,1,4
 15 3      H=HA(J)
 16        NA(J) = I1+1
 17        HA(J) = HB(J)
 18        IF(HB(NEXT) - H) 1,1,4
 19 4      HB(NEXT) = H
 20 1      CONTINUE
 21        DO 5 J=1,I1
 22        NEXT= NA(J)
 23        IF(HA(J)-HA(NEXT)) 5,6,6
 24 6      NA(J) = I1+1
 25 5      CONTINUE
 26        RETURN
 27        END
```

```
  1        FUNCTION SLINK3(NA,HA,NOBJ)
  2 C    * SLINK3 * C'EST UNE REAL*4 FUNCTION DONT LA VALEUR EST  LA
  3 C             DISTORSION IMPOSEE PAR L'ULTRAMETRIQUE : SI LES DONNEES DE DEPART
  4 C       SATISFONT DEJA L'EGALITE ULTRAMETRIQUE (C'EST-A-DIRE CORRESPONDANT A UN
  5 C           DENDROGRAMME, ALORS SLINK3 = 0
  6        INTEGER*2 NA(1)
  7        INTEGER*2 HA(1)
  8        NOBJ1 = NOBJ - 1
  9        SLINK3 = 0.0
 10        DO 1 I= 1,NOBJ1
 11        DO 2 J=1,I
 12        N1 = I+1-J
 13        IF(HA(N1)-HA(I)) 2,2,3
 14 2      CONTINUE
 15        N1=0
 16 3      I1=I+1
 17        DO 4 J=I1,NOBJ
 18        IF(HA(J)-HA(I)) 4,1,1
 19 4      CONTINUE
 20 1      SLINK3 = SLINK3 + FLOAT((I-N1)*(J-I)*HA(I))
 21        RETURN
 22        END
```

```
  1          SUBROUTINE DENDR(N,C,D,DMIN,DMAX,LLL,MOTCLE,ISW)
  2 C       * DENDR * ROUTINE D'IMPRESSION DU DENDROGRAMME
  3 C           C(I) CONTIENT LE NO DE L'OBJET A IMPRIMER  EN LIGNE NO I
  4 C           D(I) DONNE LA HAUTEUR CORRESPONDANTE DU DENDROGRAMME
  5 C           LE SWITCH ISW PERMET DE DEMANDER L'IMPRESSION
  6 C                DU MOT CLE SI ON ESSAIME DES MOTS CLES
  7 C                DU NO DE DOCUMENT SI ON ESSAIME DES DOCUMENTS
  8 C           N : NOMBRE D'OBJETS
  9 C           DMIN,DMAX  DONNENT L'ECHELLE PERMETTANT DE CONSTRUIRE LE DENDROGRAMME
 10 C          LLL INDIQUE LA HAUTEUR DU DESSIN SUR LE PAPIER
 11          INTEGER*4 DMAX,DMIN
 12          LOGICAL*1 MOTCLE(1)
 13          INTEGER*4 LINE(106)
 14          INTEGER*2 C(1),D(1)
 15          DATA IBLANK/1H /,ICHAR/1H*/
 16          LL=LLL
 17          IF(LL.GT.101) LL = 101
```

```
      DO 110 I=1,LL
110   LINE(I) = IBLANK
      PRINT 150,DMIN,DMAX
150   FORMAT(1H1,'SINGLE LINK DENDROGRAM',5X,'SCALE',I4,'--',I4//)
      FACTR = FLOAT(LL)/(DMAX-DMIN)
      NM1 = N-1
      DO 270 I=1,NM1
210   INDX = (C(I) - DMIN) * FACTR + 1
      IF(INDX.LE. 1)  INDX = 1
      IF(INDX .GT. LL )  INDX = LL
      DO 220 J=1,INDX
220    LINE(J) = ICHAR
      ASSIGN 232 TO JULES
73    GO TO (50,51),ISW
50    IK=(C(I)-1)*20 + 1
      JK=IK + 19
      PRINT 230,(40TCLE(J),J=IK,JK),(LINE(J),J=1,LL)
230   FORMAT(2X,20A1,2H--,107A1)
      GO TO 83
51    PRINT 231,C(I),(LINE(J),J=1,LL)
231   FORMAT(11X,'DOC NO',I5,2H--,107A1)
83    GO TO JULES,(232,233)
232   IF(INDX.EQ.1) GO TO 250
      INDX = INDX-1
      DO 240 J=1,INDX
240   LINE(J) = IBLANK
250   PRINT 260,(LINE(J),J=1,LL)
260   FORMAT(24X,101A1 )
270   CONTINUE
      LL=LL+5
      DO 280 J=1,LL
280   LINE(J) = ICHAR
      ASSIGN 233 TO JULES
      GO TO 73
233   CONTINUE
      RETURN
      END

      SUBROUTINE VECNOD(BRO,SON,ICARDI,LENGTH,LENR,NOFICH  )
C
C   * VECNOD * CONSTRUIT DANS ZONB LE VECTEUR REPRESENTANT LE NOEUD
C                                                  INTERMEDIAIRE
C       NBFILS MEMORISE LE NOMBRE DE NOEUDS  TERMINAUX ADMETTANT NODE COMME RACI-
C       NE DU SOUS GRAPHE PARTIEL CORRESPONDANT
C       POUR LA LECTURE DES ENREGISTREMENTS DES FILS DE NODE IL FAUT DISTINGUER
C          2 SUIVANT QUE LES FILS SONT TERMINAUX OU NON
C       ICARDI REPRESENTE LE NOMBRE D'OBJETS A ESSAIMER
C       LENGTH REPRESENTE LE NOMBRE DE NOEUDS (TERMINAUX OU NON ) DE
C                  L'ARBORESCENCE
      INTEGER*2 BRO(1),SON(1),ZONA(250),ZONB(250)
      NODE=ICARDI
11    NODE = NODE + 1
      IF(NODE-LENGTH)1,1,2
1     NBSON=0
      DO 10 L=1,LENR
10    ZONB(L)  = 0
      J = SON(NODE)
8     IF(SON(J)) 4,4,5
4     ID=J
      READ(NOFICH'ID)(ZONA(L),L=1,LENR)
      NBSON = NBSON +1
      GO TO 6
5     ID=J
      READ(NOFICH'ID)(NBFILS,(ZONA(L),L=1,LENR))
      NBSON = NBSON + NBFILS
6     DO 7 L=1,LENR
7     ZONB(L) = ZONB(L) + ZONA(L)
      J=BRO(J)
      IF(J) 9,9,8
9     ID=NODE
      WRITE(NOFICH'ID) (NBSON,(ZONB(L),L=1,LENR))
      GO TO 11
2     RETURN
      END
```

```
      SUBROUTINE EXLINK(BRO,SON,LAMBDA,PISOM,NEWNAM,NUM,ICARDI,
     1 LENGTH,LZUT  )
C    * EXLINK * BUT : CONSTRUIRE LES VECTEURS BRO ET SON A PARTIR DE LA
C           REPRESENTATION PAR POINTEURS (PISOM,LAMBDA )
C            LE FONCTIONNEMENT DE L'ALGORITHME EST EXPLIQUE AU CHAP IV DU
C                                                    MEMOIRE
C       IL Y A 2 SEQUENCES A CONSIDERER
C               A) CREATION D'UN NOUVEAU NOEUD DE L'ARBORESCENCE
C               B) INSERTION D'UN NOEUD COMME FRERE DE 2 NOEUDS DEJA REPRIS
C                   UN TEL NOEUD EST TOUJOURS INSERE EN 2EME POSITION
C           LE FAIT QUE L'ON PARCOURT UNE LISTE TRIEE ASSURE QUE TOUS LES FILS
C            EXISTENT (NOEUDS TERMINAUX OU NOEUDS INTERMEDIAIRES) LORSQU'ON
C            DECIDE DE CREER LE PERE
C
      INTEGER*2 PISOM(1),NEWNAM(1),NUM(1),BRO(1),SON(1)
      INTEGER*2 LAMBDA(1)
      LENGTH=ICARDI
      DO 1 J=1,1000
1     NEWNAM(J) = J
      CALL MINUS(PISOM,LAMBDA,NUM,ICARDI)
      IMIN=1
20    IF(IMIN-ICARDI) 10,11,11
10    LEVMIN = LAMBDA(IMIN)
      I=IMIN
      ASSIGN 14 TO IBACK
14    IF(LAMBDA(I)-LEVMIN)12,12,13
13    IMAX = I-1
      ASSIGN 23 TO IBACK
23    IF(I-ICARDI) 12,12,25
25    I=IMIN
18    IAUX=NUM(I)
      JAUX=PISOM(I)
      IF(BRO(JAUX)) 15,16,15
16    LENGTH = LENGTH+1
      SON(LENGTH)  = IAUX
      BRO(IAUX) = JAUX
      BRO(JAUX) = -LENGTH
      NEWNAM(JAUX) = LENGTH
      GO TO 17
15    INSERT = SON(-BRO(JAUX))
      MEMOR = BRO(INSERT)
      BRO(INSERT) = IAUX
      BRO(IAUX) = MEMOR
17    I=I+1
      IF(I-IMAX) 18,18,19
19    IMIN= IMAX + 1
      GO TO 20
11    LZUT=LENGTH
      RETURN
12    NUM(I) = NEWNAM(NUM(I))
      PISOM(I) = NEWNAM(PISOM(I))
      I=I+1
      GO TO IBACK,(14,23)
      END

      SUBROUTINE MINUS(PISOM,LAMBDA,NUM,ICARDI)
C    * MINUS * BUT : TRIER LES VECTEURS PISOM ET LAMBDA SUIVANT LES VALEURS CROIS
C              ANTES
C             LE VECTEUR NUM MEMORISE LES PERMUTATIONS QUI ONT ETE REALISEES
C              ICARDI INDIQUE LE NOMBRE D'OBJETS A ESSAIMER
      INTEGER*2 PISOM(1),NUM(1)
      INTEGER*2 LAMBDA(1)
      INTEGER*4 AUX
      DO 1 J=1,ICARDI
1     NUM(J) = J
      J=1
6     AUX = LAMBDA(J)
      IAUX = PISOM(J)
      INUM = NUM(J)
      I = J+1
4     IF(AUX-LAMBDA(I)) 2,2,3
3     LAMBDA(J) = LAMBDA(I)
      PISOM(J)=PISOM(I)
```

```
19        NUM(J) = NUM(I)
20        LAMBDA(I) = AUX
21        PISOM(I) = IAUX
22        NUM(I) = INUM
23        AUX = LAMBDA(J)
24        IAUX = PISOM(J)
25        INUM = NUM(J)
26 2      I=I+1
27        IF(I-ICARDI) 4,4,5
28 5      J=J+1
29        IF(J-ICARDI + 1) 6,6,7
30 7      RETURN
31        END
```

```
 1          SUBROUTINE COSLNK(ZONA,ZONB,ZOUT,LENR,NLIG,NOFICH  )
 2 C     * COSLNK * BUT :   CONSTRUIRE LA LIGNE I DE LA MATRICE TRIANGULAIRE
 3 C             INFERIEURE DES DC .
 4 C             ZONA ET ZONB SONT 2 ZONES DE TRAVAIL
 5 C             ZOUT EST LE VECTEUR RESULTAT CORRESPONDANT A LA LIGNE I
 6 C           LA VALEUR DU DC EST COMPRISE ENTRE 1 ET 20
 7          INTEGER*2 ZONA(1),ZONB(1),ZOUT(1)
 8          INTEGER*4 XX,XY,YY
 9          ID=NLIG
10          READ(NOFICH'ID) (ZONA(L),L=1,LENR)
11          N=0
12 8        N=N+1
13          IF(N-NLIG+1)  1,1,2
14 1        ID=N
15          READ(NOFICH'ID)(ZONB(L),L=1,LENR)
16          XY=0
17          XX=0
18          YY=0
19          I=1
20 6        XY=XY + ZONA(I) * ZONB(I)
21          XX = XX + ZONA(I)*ZONA(I)
22          YY = YY + ZONB(I) * ZONB(I)
23          I=I+1
24          IF(I-LENR) 6,6,7
25 7        IF((XX.EQ.0).OR.(YY.EQ.0)) GO TO 89
26          CORARI = XY/SQRT(FLOAT(XX*YY))
27           ZOUT(N)=20-INT((CORARI*20.)+0.5     )
28          GO TO 90
29 89       ZOUT(N) = 10
30 90       CONTINUE
31          GO TO 8
32 2        RETURN
33          END
```

```
 1        SUBROUTINE DEQUES(BRO,SON,NBQUES,RACINE,MOTCLE,NBTERM,X,Y,
 2       1 NFICH3,NFICH4,NFICH6,NFICH5     )
 3 C     * DEQUES * LE MODULE PERMET DE TRAITER UN NOMBRE QUELCONQUE DE QUESTIONS
 4 C         CHAQUE QUESTION EST IMPRIMEE    .  BSTRA LANCE LE RETIEVAL
 5        INTEGER*4 RACINE
 6        INTEGER*2 BRO(1),SON(1)
 7        LOGICAL*1 MOTCLE(1)
 8        INTEGER*2 COL(25),DOCTER(250)
 9        NOQUES=1
10 34     DO 27 KI=1,25
11 27     COL(KI) = 0
12 C                        * DEPOULLEMENT DES QUESTIONS
13        READ 21,JEND,(COL(K),K=1,25)
14 21     FORMAT(I5,25I3)
15        IF(JEND) 35,11,11
16 11     DO 22 L=1,NBTERM
17 22     DOCTER(L) =0
18        KI=1
19 24     J=COL(KI)
20        IF(J) 26,26,23
21 23     DOCTER(J) =1
22        KI=KI+1
23        IF(KI-25) 24,24,26
24 26     PRINT 60
25 60        FORMAT(1HO/1HO/1HO,T30,'** ENONCE DE LA QUESTION')
26        CALL ENUMER(MOTCLE,COL,KI-1,NOQUES,NBTERM,2,NFICH3,NFICH4,NFICH6)
```

```
27        PRINT 61
28 61     FORMAT(T10,'DOCUMENTS REPONDANT A LA QUESTION  '   )
29        CALL CHRONO(1)
30        CALL BSTRA(SON,BRO,RACINE,X,Y,DOCTER,NBTERM,NFICH4,NFICH5    )
31        CALL CHRONO(2)
32        NOQUES = NOQUES + 1
33        IF(NOQUES - NBQUES ) 34,34,35
34 35     RETURN
35        READ(NFICH7'I)(HB(L),L=1,I1)
36        END

 1         SUBROUTINE EXQUES(BRO,SON,NBQUES,RACINE,MOTCLE,NBTERM,X,Y,
 2        1 NFICH3,NFICH4,NFICH6,NFICH5,NBCLUS ,NPARAM      )
 3 C      * EXQUES * CONSTRUIT UNE QUESTION COMPLETEE PAR THESAURUS
 4 C              LE MODULE PERMET DE TRAITER 25 QUESTIONS AU MAXIMUM
 5 C              TOUTES LES QUESTIONS SONT LUES ET STOCKEES DANS LES 25
 6 C                              PREMIERES LIGNES DE LA MATRICE G
 7 C              ENSUITE , CHAQUE ESSAI DE MOTS CLES (STOCKE SUR NFICH6) EST AMENE
 8 C                              EN MEMOIRE CENTRALE ET EST COMPARE AVEC TOUTES LES
 9 C                              QUESTIONS : CECI LIMITE LE NOMBRE D'ACCES DISQUE
10 C              LE DERNIER PARAMETRE TANSMIS FIXE LA PRIORITE D'UN TERME DU
11 C                              L'UTILISATEUR PAR RAPPORT AUX TERMES AJOUTES PAR
12 C                              THESAURUS
13 C              LA ROUTINE BSTRA EST ENFIN APPELEE POUR LANCER LE RETRIEVAL
14         INTEGER*2 G(50,1000),IAUX(1000),ICOL(30)
15            LOGICAL*1 MOTCLE(1)
16            INTEGER*2 BRO(1),SON(1)
17            INTEGER*4 RACINE
18         INTEGER*2 DOCTER(1000)
19         COMMON/ZONE/G
20         DO 22 I=1,50
21         DO 22 J=1,NBTERM
22 22      G(I,J)=0
23         DO 29 I=1,30
24 29      ICOL(I) = 0
25         NOQUES=0
26 C                          * DEPOULLEMENT DES QUESTIONS
27 6       READ 25,INUM,(ICOL(K),K=1,25)
28 25      FORMAT(I5,25I3)
29         IF(INUM) 4,5,5
30 5       NOQUES = NOQUES+1
31         NOQ25=NOQUES+25
32         K=1
33 8       J=ICOL(K)
34         IF(J) 17,17,7
35 7       G(NOQUES,J) =1
36         G(NOQ25,J) = 1
37         K=K+1
38         GO TO 8
39 17      CALL ENUMER(MOTCLE,ICOL,K-1,NOQUES,NBTERM,2,NFICH3,NFICH4,NFICH6)
40         GO TO 6
41 4       IF(NOQUES) 19,19,20
42 20      NBQUES=NOQUES
43         NOCLUS=1
44 14      READ(NFICH6'NOCLUS)(IAUX(L),L=1,NBTERM)
45         NOQUES=1
46 12      NOQ25=NOQUES+25
47         DO 9 J=1,NBTERM
48         IF(G(NOQUES,J)+IAUX(J) -2) 9,10,10
49 9       CONTINUE
50         GO TO 32
51 10      DO 11 J=1,NBTERM
52 11      G(NOQ25,J)=G(NOQ25,J) + IAUX(J)
53 32      NOQUES=NOQUES+1
54         IF(NOQUES-NBQUES) 12,12,13
55 13      NOCLUS=NOCLUS+1
56         IF(NOCLUS-NBCLUS) 14,14,15
57 15      DO 40 NOQUES=1,NBQUES
58         NOQ25 = NOQUES+25
59         K=0
60            DO 69 J=1,NBTERM
61 69         DOCTER(J)=0
62         DO 41 J=1,NBTERM
63          IF(G(NOQUES,J))70,70,71
```

```
64 70     IF(G(NOQ25,J))41,41,72
65 72     DOCTER(J) =1
66        GO TO 43
67 71          DOCTER(J) = NPARAM
68 43     K=K+1
69        IAUX(K) =J
70 41     CONTINUE
71        CALL ENUMER(MOTCLE,IAUX,K,NOQUES,NBTERM,2,NFICH3,NFICH4,NFICH6)
72        PRINT 61,NOQUES
73 61     FORMAT(1HO/1HO,T10, ' DOCUMENTS REPONDANT A LA QUESTION ',I4)
74        CALL CHRONO(1)
75        CALL BSTRA(SON,BRO,RACINE,X,Y,DOCTER,NBTERM,NFICH4,NFICH5    )
76        CALL CHRONO(2)
77 40     CONTINUE
78 19     RETURN
79        END

 1        SUBROUTINE LISDOC(BRO,SON,NODE,Q,NBTERM,NFICH4,NFICH5  )
 2 C       * LISDOC * BUT ENUMERER TOUS LES NOEUDS TERMINAUX DU SOUS-GRAPHE PARTIEL
 3 C                    ET AYANT NODE COMME RACINE
 4 C              UN STACK MEMORISE LES NOEUDS INTERMEDIAIRES
 5 C              LA RECHERCHE DES NOEUDS TERMINAUX SE FAIT PAR UNE METHODE TOP-DOWN
 6 C                 ET LEFT-RIGHT
 7 C              NOUS DONNONS EN PLUS  LA VALEUR  DE LA CORRELATION SINUSOIDALE
 8 C                 ENTRE LE DOCUMENT ENUMERE ET LA QUESTION Q
 9        INTEGER*2 Q(1)
10        INTEGER*2 ZONA(250)
11        LOGICAL*1 CARTE(80)
12        INTEGER*2 BRO(1),SON(1)
13        INTEGER*4 STACK(250)
14        INTEGER*4 HPIL,NODE
15        INTEGER*4 XX,YY,XY
16        ASSIGN 11 TO IBACK
17        IAUX=SON(NODE)
18        IF(IAUX) 1,1,2
19 2      STACK(1) = IAUX
20        HPIL=1
21 6      IAUX=SON(IAUX)
22        IF(IAUX) 3,3,4
23 4      HPIL  = HPIL+1
24        STACK(HPIL) = IAUX
25        GO TO 6
26 3      ID = STACK(HPIL)
27 20     READ(NFICH4'ID)(CARTE(L),L=1,80)
28        READ(NFICH5'ID)(ZONA(L),L=1,NBTERM)
29        XX=0
30        XY=0
31        YY=0
32        DO 73 IK=1,NBTERM
33        XY=XY + ZONA(IK)*Q(IK)
34        XX=XX + ZONA(IK) * ZONA(IK)
35 73     YY = YY + Q(IK) * Q(IK)
36        IF((XX.EQ.0).OR.(YY.EQ.0)) GO TO 89
37        COR = XY/SQRT(FLOAT(XX*YY))
38        GO TO 90
39 89     COR=0.0
40 90     PRINT 800,(CARTE(L),L=1,80),COR
41 800    FORMAT(T10,80A1,' * CORRELATION * ',F5.3
42        GO TO IBACK,(11,10)
43 11     IAUX = BRO(STACK(HPIL))
44        IF(IAUX) 7,7,8
45 8      STACK(HPIL) = IAUX
46        GO TO 6
47 7      HPIL = HPIL - 1
48        IF(HPIL) 10,10,11
49 1      ASSIGN 10 TO IBACK
50        ID=NODE
51        GO TO 20
52 10     RETURN
53        END
```

```
      SUBROUTINE BSTRA(SON,BRO,RACINE,X,Y,Q,NBTERM,NFICH4,NFICH5    )
C     * BSTRA* (BROAD SEARCH STRATEGY )
C          LA ROUTINE DIRIGE LE RETRIEVAL COMME INDIQUE DANS LE CHAPITRE IV
C                                                          DU MEMOIRE
C          ON COMMENCE PAR ETUDIER LE 1ER FILS DE LA RACINE
C             CE FILS EST MEMORISE DANS UN STACK AFIN DE POUVOIR ETUDIER SES
C                                                                  FRERES
C             AU FUR ET A MESURE QUE L'ON DESCEND DANS L'ARBORESCENCE , LE STACK
C             SE REMPLIT: LA HAUTEUR DU STACK DIMINUE LORSQU'ON EXAMINE TOUS LES
C              FILS D'UN NOEUD DONNE ( CONDITION DETECTEE PAR UNE VALEUR
C                              NEGATIVE      AU SOMMET DU STACK
C             LORSQU'ON A TROVE UN SOMMET REPONDANT AUX CONDITIONS IMPOSEES,ON
C             IMPRIME TOUS LES SOMMETS TERMINAUX CORRESPONDANTS (CALL LISDOC )
      INTEGER*4 RACINE
      REAL*4 X,Y,F,G
      INTEGER*2 BRO(1),SON(1)
      INTEGER*4 STACK(250)
      INTEGER*2 Q(1)
      I=1
      IAUX= SON(RACINE)
10    STACK(I) = IAUX
      CALL FGCQ(Q,IAUX,NBTERM,NFICH5,F,G,SON   )
      IF(F-X) 4,5,5
5     IF(G-Y) 6,7,7
6     IAUX= SON(STACK(I))
      IF(IAUX) 7,7,8
8     I=I+1
      GO TO 10
7     CALL LISDOC(BRO,SON,STACK(I),Q,NBTERM,NFICH4,NFICH5   )
4     IAUX = BRO(STACK(I))
      IF(IAUX) 9,9,10
9     I=I-1
      IF(I) 11,11,4
11    RETURN
      END
```

```
      SUBROUTINE FGCQ(Q,NODE,NBTERM,NOFICH,FCQ,GCQ,SON   )
C     * FGCQ * CONSTRUIT POUR LE NOEUD NODE ET LA QUESTION Q LES  VALEURS
C                              DES FONCTIONS F(Q,N) ET G(Q,N) UTILISEES DANS BSTRA
C        CARDQ ET CARDC SONT LES POIDS CORRESPONDANT A Q ET N
C        LE VECTEUR REP STOCKE LA REPRESENTATION DU NOEUD NODE
      INTEGER*2 SON(1)
      INTEGER*4 TE,CARDC,CARDQ
      REAL*4 FCQ,GCQ
      INTEGER*2 REP(250),Q(1)
      TE=0
      CARDQ = 0
      IF(SON(NODE)) 1,1,2
1     ID=NODE
      READ(NOFICH'ID)(REP(L),L=1,NBTERM)
      CARDC=1
      GO TO 3
2     ID=NODE
      READ(NOFICH'ID) (CARDC,(REP(L),L=1,NBTERM )     )
3     J=1
6     IF(Q(J)) 4,4,5
5     TE=TE+REP(J)
      CARDQ = CARDQ + 1
4     J=J+1
      IF(J-NBTERM) 6,6,7
7     FCQ=FLOAT(TE)/CARDQ
      GCQ = FLOAT(TE)/CARDC
      RETURN
      END
```

```
      SUBROUTINE CHRONO(IARG)
C     * CHRONO * COMPTE LE TEMPPS CPU ENTRE 2 POINTS QUELCONQUES DU PROG.
C              SI IARG = 1 INITIALISATION DU CHRONO
C              SI IARG = 2 IMPRESSION DE LA VALEUR AFFICHEE PAR LE CHRONO
C              ON COMMENCE PAR ETUDIER LE 1ER FILS DE LA RACINE
      GO TO (1000,2000),IARG
1000  PRINT 60
60    FORMAT(1HO,T10,10(1H*),' CHRONO IS INITIALIZED ',10(1H*))
      CALL TIMEX(IANS,IMOIS,IJOUR,IHEURE,IMINUT,ISEC,IDM)
      IBEGIN = IDM
      RETURN
2000  CALL TIMEX(IANS,IMOIS,IJOUR,IHEURE,IMINUT,ISEC,IDM  )
      ITIME= (IDM-IBEGIN)/10
      PRINT 61,ITIME
61    FORMAT(1HO/T10,10(1H*),' CHRONO IS STOPPED AT ',I10,'  MSEC  ',
     1  10(1H*))
      RETURN
      END
```

BIBLIOGRAPHIE

AITCHISON,S.(1970)The thesaurofacet:A multipurpose retrieval language tool,J.of Docum.26 3,pp 187-203

ANDERBERG,M.(1973)Cluster Analysis for applications,Academic Press

AUGUSTSON,S.,MINKER,S.(1970)An Analysis of some graph theoritical cluster techniques,JACM,17 pp 571-588

BALL,G.(1965)Data Analysis in the Social Sciences:What about the details,Proc.Fall Joint Computer Conference,27 pp 533-559

BERMAN,G.,COLIJN,A.(1974)A modified list technique allowing binary search,JACM,21,2,pp 227-232

BONNER,R.(1964)One some clustering techniques,IBM J.Res and develop., 8,1,pp 22-32

BORKO,H.,BERNICK,M.(1963)Automatic document classification,JACM,10, pp 151-162

BOUROCHE,J.M.,CURVALLE,B.(1974)La recherche documentaire par voisinage, RAIRO,5,1,pp65-96

BRAUEN,T.L.(1971)Document vector modification:voir Salton 71 Ch.24

BREUER,M.(1972)Design automation of Digital Systems,vol 1,Prentice Hall

BRON,C.,KERBOSCH,J.(1973)Finding all cliques of an undirected graph, Algorithm 457,CACM,16,9,pp575-577

BURKHARD,N.,KELLER,R.(1973)Some approaches to Best match file searching, CACM,16,pp 230-236

CAGAN,C.(1970)A highly associative document retrieval system,J.ASIS. pp330-337

CASEY,R.G.(1973)Design of tree structures for efficient querying,CACM, 16,pp 549-556

CHIEN,R.,MARK,E.(1974)A document storage method based on polarized distance,JACM,21,3,pp 233-245

CLEVERDON,C.(1972)On the inverse relationship of recall and precision, J.of Docum 28,3,pp 195-201

COOPER,W.S.(1971)A definition of relevance for information,I.S.R.,7, pp 19-37

CORMACK,R.(1971)A review of classification,J.of Royal Statistical Society A,33,pp 321-367

CROS,R.,GARDIN,S.,LEVY,F.(1964)L'automatisation des recherches documentaires:un modèle général,le Syntol,Gauthier - Villars

DATTOLA,R.(1969)A fast algorithm for automatic classification,J.of Library Automation,2/1,pp 31-48

DILLON,M.(1974)An experiment in superficial indexing,ISR,10,pp 63-71

EDMUNDSON,N.(1969)New methods in automatic extracting,JACM,16,2,PP 297-314

FARRADANE,S.,RUSSEL,S.YATES-MERCER,P.(1973)Problems in information retrieval:Logical jumps in the expression of information,ISR, 9,pp 65-77

FICHEFET(1974)Théorie des graphes et applications,FNDP,Institut d'Informatique,NAMUR

FISHER,L.,VANNESS,S.(1971)Admissible clustering procedures,Biometrika, 58,1,pp91-104

GOTLIEB,C.,KUMAR,S.(1968)Semantic clustering of index terms,JACM,15,4 pp 493-513

GOWER,J.(1969)What is cluster analysis?(abstracts)Biometrics,25,pp609-610

GREEN,FRANK,ROBINSON(1967)Cluster analysis in test market selection, Man.Sci.,13,8,pp387-400

HARRISON,I.(1968)Cluster analysis,Metra,7,pp513-528

HARTIGAN(1967)Representation of similarity trees by trees,JASIS,62, pp 1140-1158

HSIAO,D.,HARARY,F.,A formal system for information retrieval from files, CACM,13,2,pp 67-73

HUANG,S.(1973) A note on information organisation and storage,CACM,16, 7,pp 406-410

JACKSON(1971) Classification,relevance and information retrieval, Advances in computers,11,pp 59-125

JACQUESSON,A.,SCHIEBER,W.(1973)The term association analysis on a large file of bibliographic data,using a highly controlled indexing vocabulary,ISR,9,pp 85-94

JARDINE,C.JARDINE,N.,SIBSON,R.(1967)The structure and construction of taxonomic hierarchies,Mathematical Biosciences,1,pp 173-179

JARDINE,N.(1970)Algorithms,methods and models in the simplification of complex data,the Comp.Journal,13,1,pp116-117

JARDINE,N.,SIBSON,R.(1968) The construction of hierarchic and non hierarchic classifications,the Comp.J.,11,pp 177-183

JARDINE,N.,SIBSON,R.(1968) A model for taxonomy,Math.Biosc.,2,pp 465-482

JARDINE,N.,SIBSON,R.(1971) Choices of methods for automatic classifications,the Comp.J.,14,4,pp 404-406

JARDINE,N.,SIBSON,R.(1971) Mathematical taxonomy,Wiley,London

JARDINE,N.,VANRIJSBERGEN,C.(1971) The use of hierarchic clustering information retrieval,ISR,7,pp 217-240

JENSEN,R.(1968) A dynamic programming algorithm of cluster analysis OP.Res. pp 1034-1056

JOHNSON,LAFUENTE(1970) A controlled single-pass classification algorithm with application to multilevel clustering Section XII, Dept of Computer Science,Cornell 26

KERCHNER,M.(1971) Dynamic document processing in clustered collections Scientific Rep. ISR-19,Dept of Computer Science,Cornell

KNUTH,D.(1969) The art of computer programming,Vol I. Addison-Wesley

LERMAN,I.(1970) Les bases de la classification automatique, Gauthiers-Villars.

LESK,M.E.(1969) Word-word associations in document retrieval systems, Amer.docum. 20,pp 27-38

LESK,M.,SALTON,G.(1969) Interactive search and retrieval methods using automatic information displays,Proc. AFIPS, Spring Joint Computer Conference, pp 435-446.

LESSER,V.(1966) A modified two level search algorithm using request clustering,Scientific Report,ISR,11,Dept of Computer Science, Cornell U.

McNAUGHTON-SMITH(1963) The classification of individuals by possession of attributes associated with a criterion. Biometrics 19, pp 364-366.

MARON,M.,KUHNS,J.(1960) On relevance,probabilistic indexing and information retrieval,JACM,1,pp 216-244

MEETHAM,A.(1966) Graph separablility and Word grouping. Proceeding - ACM.National meeting,pp 513-514

MILLER,W.(1971) A probabilistic search strategy for MEDLARS. J. of Docum. 27,4,pp 254-266

MINKER,S.,WILSON,G.,ZIMMERMAN,B.(1972) An evaluation of query - expansion by the addition of clustered terms for a document retrieval system ISR,8,pp 329-348

MOON,J.,MOSER,C.(1965) On cliques in graphs,Israël J. Math.3,pp 23-28

MULLIGAN,G.,CORNEIL,D.(1972) Corrections to Bierstone's algorithm for generating cliques,JACM,19,pp 244-247

MURRAY,D.M(1972) Document retrieval based on clustered files.Report ISR -20 to National Science of Foundations,Cornell University

NAULLEAU,D.(1973) Thesaurus conversationnel - Conception et réalisation d'un outil aidant à l'indexation et à la formulation de questions,thèse présentée à l'université de Paris VI pour l'obtention du doctorat 3ème cycle

NEEDHAM,R.M.(1965) Applications of the theory of clumps,Mech. Transl. Comput.linguistics,8,pp 113-127

OGILVIE,J.(1969) The distribution of number and size of connected components in random graphs of medium size.Information Processing 68 - North Holland Publishing Company,pp 1527-1530

PIKE,M.,PETO(1970) Some clustering problems in medical Research Biometrics,26,p 356(abstract)

PIKE,M.,SMITH(1968) Disease clustering:a generalisation of Knox's approach to the detection of space-time interactions,

Biometrics,24,pp541-556

RESNIKOFF,DOLBY(1971) Access study of information storage and retrieval with emphasis on library information systems. V.S. Dept. of health,education and welfare,U.S.A.

ROGERS,D,TANIMOTO,T.(1960) A computer program for classifying plants, Science,132,pp 1115-1118

ROHLF,F.(1973) Hierarchical clustering using the minimum spanning tree, Algorithm 76,the Comput.J.16,pp 93-95

ROHLF,F.(1974) A dendrogram plot,Algorithm 81,THE Comput.J.17,pp 89-91

SALTON,G.(1963) Associative document retrieval techniques using bibliographic information,JACM,10,4,pp 440-457

SALTON,G.(1966) Information,dissemination and automatic Information systems.Procedings of the IEEE,54,pp 1663-1678

SALTON,G.(1968) Search strategy and the optimization of retrieval effectiveness,Mechanized information storage,Retrieval and Dissemination,pp 73-107

SALTON,G.(1968) Automatic information organization and retrieval, MAC GRAWHILL

SALTON,G.(1971) The SMART Retrieval system,Experiments in Automatic document processing. Prentice Hall

SALTON,G.(1971) The performance of interactive information retrieval, Information Processing Letters,1,pp 35-41

SALTON,G.(1972) A new comparaison between conventional indexing(MEDLARS) and automatic text processing(SMART) J.ASIS,pp 75-82

SALTON,G.(1972) Experiments in automatic thesaurus construction for information retrieval,Information Processing,71,North Holland Publishing Company,pp115-123

SALTON,G.(1972) Dynamic document processing,CACM,15,pp 658-678

SALTON,G.(1973) Recent studies in automatic text analysis and document retrieval,JACM,20,pp 258-278

SALTON,G.,LESK,M.(1969) Relevance assessments and retrieval system evaluation,ISR,4,pp 343-359

SCHNEIDER,M.(1973) Méthodes pour recenser toutes les cliques maximales et (-) maximales d'un graphe,RAIRO,5,3,pp 21-33

SCOTT,A.,SYMMONS,M.(1971) Clustering methods based on likelihood ratio criteria,Biometrics,27,pp 387-397

SHEPHERD,M.S.,WILMOTT,A. Cluster analysis on the ATLAS computer,11, the Comp.J. p 57

SIBSON,R.(1972) Order invariant methods for data analysis,J.Royal Stat. Society,34,pp 3LL-349

SIBSON,R.(1973).SLINK:An optimally efficient algorithm for the single-link cluster method 16,pp 30-34

SPARCK-JONES,K.(1969) Automatic term classification and information retrieval,Information Processing 68,pp 1290-1295

SPARCK-JONES(1971) Automatic keyword classification for information retrieval, Butterworths, London

SPARCK-JONES,K.(1972) Statistical interpretation of term specificity and its application in retrieval, J.Docum.28, pp 11-21

SPARCK-JONES(1973) Index term weighting, ISR, 9, pp 619-633

SPARCK-JONES(1973) Collection properties influencing automatic term classification performance, ISR, 9, pp 499-513

SPARCK-JONES,K., BARBER,E.(1971) What makes an automatic keyword classification effective? JASIS, pp 166-175

SPARCK-JONES,K., JACKSON,D.M.(1967) Current approaches to classification and clump finding at the Cambridge language resea rch unit, the Comp.J.10,1, pp 29-37

SPARCK-JONES,K., JACKSON,D.(1970) The use of automatically obtained keyword classifications for information retrieval ISR, 5, pp175-201

SPARCK-JONES,K., NEEDHAM(1968) Automatic term Classifications and retrieval, ISR, 4, pp 91-100

SRINIVASAN,V., THOMPSON,G.(1972) Accelerated algorithms for labeling and relabeling of trees, with applications to distributions problems, JACM, 19, pp 712-726

STANFELD,E.(1972) Practical aspects of doubly chained trees for retrieval, JACM, 19, 3, pp 425-436

STILES,H.(1961) The Association factor in information retrieval, JACM, 8, 2, pp271-279.

SWANSON,D.R.(1963) Interrogating a computer in natural language, Information Processing 62, North Holland Pubishing Company pp251-257

VANRIJSBERGEN,C.(1970) Aclustering algorithm -Algorithm 47- The Comp. J. 13, pp 113-115.

VANRIJSBERGEN,C.(1974) Further experiments with hierarchic clustering in document retrieval, ISR, 10, pp 1-14.

VANRIJSBERGEN,C., SPARCK-JONES,K.(1973)A test for the separation of relevant and non relevant documents in experimental retrieval collections, J.of Docum., 29, pp 251-257.

VASWANI,P.(1969) A technique for cluster emphasis and its applications to automatic indexing, Information Processing68, North Holland Publising Company pp 1300-1303.

VICHERY(1970) Techniques of information retrieval, Butterwoths, London 70

WATANABE,S(1972) A unified view of clustering algorithms, Information Processing 71, pp 149-154.

WARD,J.(1963) Hierarchical grouping to optimise an objective function, J.ASIS, 58, pp 236 -244

WILKINSON,E.(1971) Archeological seriation and the travelling salesman problem, Mathematics in the Archeological Sciences (HODSON, KENDALL, TAUTU, eds)pp 276-283